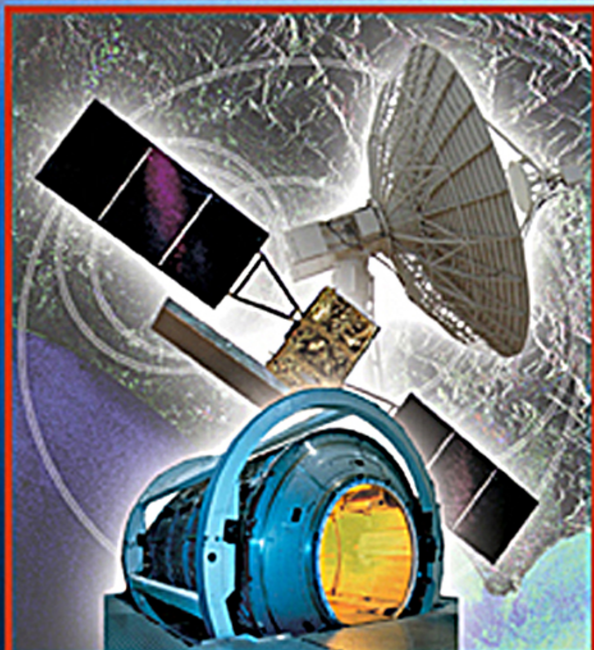
Space Operations

Mission Management, Technologies, and Current Applications

Edited by
Loredana Bruca
J. Paul Douglas
Trevor Sorensen



PROGRESS IN ASTRONAUTICS AND AERONAUTICS

Frank K. Lu, Editor-in-Chief
Volume 220

Space Operations: Mission Management, Technologies, and Current Applications

This page intentionally left blank

Space Operations: Mission Management, Technologies, and Current Applications

Edited by
Loredana Bruca
Agenzia Spaziale Italiana
Rome, Italy

J. Paul Douglas
ASRC Aerospace Corporation
Greenbelt, Maryland

Trevor Sorensen
University of Hawaii
Honolulu, Hawaii

Volume 220
PROGRESS IN
ASTRONAUTICS AND AERONAUTICS

Frank K. Lu, Editor-in-Chief
University of Texas at Arlington
Arlington, Texas

Published by the
American Institute of Aeronautics and Astronautics, Inc.
1801 Alexander Bell Drive, Reston, Virginia 20191-4344

American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia

1 2 3 4 5

ISBN10: 1-56347-919-2
ISBN13: 978-1-56347-919-9

This page intentionally left blank

Foreword

At the Ninth SpaceOps Symposium a total of 280 papers were presented and discussed in 76 subject-oriented sessions. This volume, *Space Operations: Mission Management, Technologies, and Current Applications,* presents 36 selected papers from the Symposium. The breadth and depth of the topics covered make this book an excellent addition to the AIAA Progress in Astronautics and Aeronautics Series.

The selection of 36 reviewed and updated papers published in this book was driven by their quality and relevance to the space operations community. The papers represent a cross section of three main subject areas: Spacecraft Operations, Ground Operations, and Management. The general scope of these terms, as promoted by the SpaceOps Organization, follows.

Spacecraft Operations covers the preparation and implementation of all activities to operate a space vehicle (manned and unmanned) under normal, non-nominal, and emergency conditions. This definition includes the design, production, and qualification of all means (tools, procedures, and trained personnel) to perform the task of spacecraft operations. The main challenge in this area is the cost-efficient combination of tools, degree of automation, and staffing to provide secure and reliable operations.

Ground Operations covers the preparation, qualification, and operations of a mission-dedicated ground segment and appropriate infrastructure including antennas, control centers, and communication means and interfaces. Ground operations in this context covers the design, implementation, and qualification of a particular ground segment, the provision of ground segment operations tools, provision of simulation means, ground segment operation, configuration management, and maintenance of the ground segment. The main challenge in this area is the implementation and operation of high-quality, cost-efficient, and secure space-to-ground, space-to-space, and ground-to-ground communications respecting a multitude of different standards and protocols.

Management covers all management tasks for preparing and operating a particular mission. The main talent expected is an expert level of technical understanding of spacecraft and ground operations requirements resulting in the planning and execution of all defined activities within a given schedule and within the available budget. Managers also should have a good understanding of the organizational influences on spacecraft and ground operations to minimize risks of mishaps and accidents and the behavioral aspects of human error in the conduct of the mission operations. For cooperative international missions, managerial skills in dealing with international partners and agencies and intimate knowledge of the "culture" and policy of a particular organization or agency are considered to be mandatory.

The papers selected for this volume not only are representative of the main subject areas just described but also provide a snapshot of current terminology

as it was employed throughout various presentations, round table discussions, and plenary sessions during the Symposium, sometimes with more than one meaning. This terminology is fluid and reflects progress and change within various fields of space operations. Another outcome of the Symposium, therefore, was the collection of "buzz words" that were used in an attempt to define operational expressions and standardize the language of space operations. Creation of a systematic "SpaceOps Glossary" may be the next step of this exercise, and the Symposium and publication of this book are logical springboards for this effort. The following list of words and phrases reflects the initial effort to codify the language of the space operations community. Corresponding chapter numbers are noted where glossary terms are used within this volume. Non-referenced terms can be found in the other papers presented at the Symposium.

Automation

Automation of operations is achieved by implementation of centralized telemetry, telecommand, planning, and scheduling tools requiring very little or no operator interactions (*Chapter 18*). Introduction of automation into operational systems is particularly challenging (cost-intensive). In human spaceflight, astronauts add to the complexity because of sometimes-unpredictable reactions. Introduction of increasing levels of ground-station remote monitoring and control with no impact on quality of service is being exercised. Automation of space segment through the use of robotics is being employed; end-effectors and tool exchange require much more complex operations preparation. The beneficial aspect of automation is that more autonomy onboard allows lower operations staffing in case of nominal behavior; in case of contingencies, however, deeper expertise may be required, depending also on the level of validation on the onboard automation performed during mission preparation. A rich, three-dimensional, visualization environment can contribute to a virtual operational presence in space and supports terrestrial distributed operations (*Chapter 19*). Definition and issues of an automated service within a mission operations framework (*Chapter 23*) are being investigated within the community. Automation proposals for scientific image data collection, by comparing the (changing) status of the observation object, are a means to improve the efficiency of data return.

Crew Autonomy

Because of the increasing delay times for long-distance human spaceflight, crew autonomy is mandatory with independent and automated onboard planning systems (*Chapter 21*).

Communications Technology

Space communications (space-to-ground, ground-to-space, and space-to-space) always demand higher transmission capabilities (data rates and protocols) for up- and downlink data transfer. Interesting and promising validation results of

interplanetary Ka-band link (high rate) communications usage have been presented. The European Space Agency ground-segment management is looking into Ka-band expansion as well. For interplanetary (lunar/Mars) missions, the use of IP rather than point-to-point protocols has been proposed.

Delay-Tolerant User Interface

Because of the increasing delay times for long-distance human spaceflight, proposals for delay-tolerant (scientific) user interfaces are discussed (*Chapter 21*).

Delay-Tolerant Network (DTN)

Because of the increasing delay times for long-distance human spaceflight, the methods of establishing more delay-tolerant networks are being considered (*Chapter 21*).

Formation Control and Swarms

With increasing use of smaller but multiple (identical) application satellites, the intelligent coordination and control of the formation constituents becomes more difficult than operations of a single spacecraft. Use of magnetic fields to maintain the formation coordination has been proposed. Large numbers of multiple agents within a swarm must be multiple-failure-tolerant and the agents must cooperate with each other for the achievements of their high-level goals.

International Collaboration

One important aspect of international collaboration would be the free exchange of know-how and hardware among the participants. International Traffic in Arms Regulations (ITAR) must be addressed to resolve issues of international collaboration with the United States.

Internet for Space

Internet for space to be used for long-duration human spaceflight has been suggested (*Chapter 21*).

Interoperability

The scope and dynamics of spaceflight operations are changing. More interrelated platforms require more interoperability. Universal numbering schemes, XML/UDL schemas, and messaging transport are requested. One proposal is to establish a Registry of Registries of “closed” systems with the appropriate information.

Layered Architecture

Layered architectures are very common (“rampant”) because of cost effectiveness. Software architecture relies increasingly on commercial off-the-shelf (COTS) products (*Chapter 18*).

Logistics

Radio Frequency (RF) object identification (supermarket approach) is one option to solve the increasingly complex onboard logistics problems encountered during long-duration human spaceflight.

Middleware

The term "middleware" is used for software/hardware components introduced to interconnect different, inhomogenous software programs/layers. The middleware issue and how to approach this problem (interface adaptation) is a topic that generates considerable interest with various possible solutions.

Multimission Operations

The multimission operations concept capitalizes on the same (slightly increased) set of multiskilled operators using common software/hardware/ground architecture components for more than one mission. As a result, multimission operations manpower savings reduce operations cost for specific missions. The introduction of multimission operations concepts are now seen to be taken for granted, which indicates that people have recognized the advantages. Multimission approaches are considered sufficiently mature enough for standardization.

Operations Tools

Classical operations tools are telemetry (TM), telecommand (TC), mission planning, orbit/attitude determination, and control packages. It is suggested that the tools need to not only integrate just operations content but also life-cycle cost assessments for certain operations options to be used for final management decisions. The use of not-yet-qualified operations tools in operations preparation activities should be introduced as early as possible: "test as you fly."

Public–Private Partnership (PPP)

New Zealand uses a PPP approach model very successfully to provide government services (technology, hardware, bureaucracy, government markets) to private organizations contributing innovation, private investment, and commercial markets. The application of this model has been discussed for commercial space transportation for various scenarios, such as near-Earth or lunar exploration (*Chapter 2*). Some advocate that space business could grow by sharing the risk in PPP's according to similar models.

Requirements Tracing

Requirements tracing is important for the design, implementation, qualification, and acceptance processes. Complex systems (thousands of requirements and interfaces) require automated tools. A detailed description of an advanced tracing and verification tool incorporating the definition of test cases, procedures, and status of test-case mapping based on COTS is presented. The final tracing within a verification control database (VCDB), as well as its methodology, has been demonstrated.

Return on Experience (ROE)

ROE is the transfer of experience from previous missions to new projects for enhancement of efficiency, reliability, safety. This concerns not only the staffing of new projects with experienced people (whenever possible) but also the capitalization of past experience gained in the design, development, qualification, and/or operations of a relevant mission. Current experience is well published in SpaceOps papers and presentations (*Chapter 36*), however there seems to be a transfer problem not only between successive projects but also between generations. "Lessons learned" reports are not always getting the attention they deserve. One of the major goals of the SpaceOps Symposia is to improve and spread awareness of the vast pool of operational experience within the community.

Security

Security issues in spaceflight are very complex and not very highly publicized. One facet of security is user authentication. An ESA-developed telecommand (TC) authentication scheme with a probability of discovering the key on the order of 1%, has been discussed. However this "discovering probability" is considered to be inadequate for specific ESA security requirements.

Service-Oriented Architecture

Basics of service-oriented architectures and interactions of applications, components, and infrastructure have been presented (*Chapter 10*). Some advocate that higher integration of flight and ground aspects should be *event* driven, not service driven, to improve configurability and extensibility.

Simulators

Simulators have to tackle increasingly complex problems like satellite constellations and rover operations (*Chapter 32*). Simulators have a growing important role for knowledge maintenance over long-duration missions, as well as for public education.

Software Development

AGILE programming as a new method to reduce overall software development cost in a mission control environment has been recommended. The use of complex and flexible telemetry/telecommand (TM/TC) standard packages often does not yield the expected cost savings because of high customization efforts.

SpaceOps

Currently 13 international space agencies participate in SpaceOps (the International Committee on Technical Interchange for Space Mission Operations and Ground Data Systems), established in 1992 to foster continuous technical interchange on all aspects of space-mission operations and ground-data systems,

and to promote and maintain an international community of space operations experts from agencies, academic institutions, operators, and industry.

Standards

Spaceflight standards for ground and onboard are basically driven by the International CCSDS and the ESA ECSS standards. CCSDS key words include layers, wrapability, and swapability (*Chapter 7*). Positive experiences with XTCE have been reported. Deployment of (CCSDS-) Space Link Extension (SLE) standards have made quick progress in Europe, however, outside of Europe discussion of SLE is minimal; is this as a consequence of Europe being ahead of other regions in SLE rollout? There is growing use of standardization, particularly SMP2 standard in simulators. SMP2 is due to become an ECSS standard. ESA studies the unification of ground systems in Europe (standardization of interfaces), but there is a long way to go. Basic ideas for standardizing scientific operations systems (SOS) have been presented and a proposal before CCSDS suggests establishing a working group on this subject. Multimission operations concepts and their standardization must "go in one breath." There are lots of competing standards out there, suggesting that the field is in a healthy state.

System of Systems

International participation is high in the exchange of Earth observation information for prediction, damage management, and relief-effort coordination. This could be a model for further international cooperation and implementation of the Vision for Space Exploration (VSE).

System Validation

System validation of all contributing ground subsystems is a precondition for operational qualification and is usually being performed as an independent activity. Experience has shown that early system validation testing (i.e., not all involved subsystems have the same maturity) is beneficial for ground operations.

This volume, *Space Operations: Mission Management, Technologies, and Current Applications*, is rich with examples of the type of terminology that is currently being used within our community. Creation of the "SpaceOps Glossary" will help us define and standardize the language further. If you wish to be a part of this effort, your contributions will be welcome. Please send comments to joachimkehr@opsjournal.org.

Joachim Kehr
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
German Space Operations Center (GSOC)
April 2007

Table of Contents

VIII. Moon, Mars, and Beyond

IX. Operations Experiences

This page intentionally left blank

Preface

SpaceOps is a forum founded in 1990 for providing technical interchange on all aspects of space mission operations and ground data systems among experts from space agencies, academic institutions, space operators, and industry.

The SpaceOps Organization aims at encouraging and facilitating the interchange of managerial and technical information via periodic Symposia concerning ground data systems and mission operations. Other formal and informal meetings, workshops, and the publication and dissemination of managerial and technical information have been included.

The symposia are organized on a biennial basis. They are hosted and organized by a selected participating space agency. Conference features include technical sessions, plenary sessions, poster presentations, social and networking events, industry exhibition and sponsorship opportunities. The Ninth Symposium, organized in Rome by the Italian Space Agency (ASI) on 19–23 June 2006, focused on the theme “Earth, Moon, Mars, and Beyond.”

Following the 2006 symposium, the SpaceOps Executive Committee and AIAA decided to publish in a book a selection of 36 papers reflecting the more representative subjects presented at the symposium. These papers were reviewed to assess the technical accuracy, completeness, and usefulness (no redundancy) of the information, and were also analyzed regarding clarity, logical organization, and emphasis of importance to space operations. The resulting volume is organized into nine parts referring to the conference theme and technical content, which were centered on the following topics: Mission Management, Standards, Ground Systems, Communications and Tracking, Automation, Planning Tools and Advanced Technologies, Earth Orbiting Missions, Moon, Mars, and Beyond, and Operations Experiences.

The Reviewers Committee consisted of members from the SpaceOps Organization Publication Group and Space Operations and Support Technical Committee of AIAA:

Eduardo Bergamini (INPE)
Geneviève Campan (CNES)
J. Paul Douglas (ASRC Aerospace Corp.)
Andy Dowen (NASA–JPL)
David Fuller (Macconnect)
Robert Hall (AGI)
Joachim Kehr (DLR)
Gary Kinnaman (Honeywell)
Paolo Maldari (ESA)
Mike Mattis (Maximsys)
Franz Newland (Terma)
Carrie Olsen (Mississippi State Univ.)
Norman Reese (NASA)
Dan Showalter (CSA)
Trevor Sorensen (Univ. of Hawaii)
Yoshitaka Taromaru (JAXA)

The coordinated effort of a number of people from the Reviewers Committee was instrumental in guiding the other members of the SpaceOps Editorial Board

and the authors who contributed to the publication. The support and assistance of the AIAA Books Department staff also was invaluable throughout the publication process.

Loredana Bruca
J. Paul Douglas
Trevor Sorensen
May 2007

I. Mission Management

This page intentionally left blank

Chapter 1

Cost-Effective Spacecraft Operations: Earth Observation Family of Missions Concept

P. P. Emanuelli,* N. Mardle,† and P. G. Bargellini‡

European Space Agency, Darmstadt, Germany

I. Introduction

SATELLITE operations can make up a significant percentage of the overall cost of an ESA mission, because of the length of the missions and the design and development of new systems. In the Earth Observation Missions Division at the European Space Operations Centre (ESOC), a new concept was adopted for the next generation of Earth Explorer satellites to make it more cost and resource effective. This Family of Missions concept, while not being unique, has allowed the development and implementation of the systems to be more cost effective, by reuse of overall methodology, equipment, and personnel. Starting with CryoSat and Gravity Field and Steady-State Ocean Circulation Explore (GOCE), a combined operations concept was developed which was then applied to Aeolus, Swarm, and later missions, ensuring that the most efficient use of the available resources was made. This chapter discusses the Family of Missions concept and how it has been applied in the Flight Operations Segment (FOS) of the Earth Observation (EO) Missions Division.

II. Responsibilities of the Flight Operations Segment

The FOS is responsible for operating the satellites during the various phases of its mission. To do this there are a number of activities to prepare for and then

---

*Head of Earth Observation Missions Division.

†CryoSat Spacecraft Operations Manager, Earth Observation Missions Division.

‡Aeolus Spacecraft Operations Manager, Earth Observation Missions Division.

assume the operational responsibility for the spacecraft, including: 1) operational interface requirements definition, 2) ground segment requirements definition, 3) ground segment design, integration, and validation for the following subsystems: Mission Control System (MCS) including external facilities interfaces, simulator, ground stations, communications, and Flight Dynamics Systems (FDS); 4) satellite operational database installation and validation; and 5) flight operations procedures production and validation. All of these elements are required to support the various phases of a mission, including: 1) launch and early orbit phase (LEOP), 2) commissioning, 3) routine phase, including contingency recovery operations, and, 4) decommissioning (if applicable).

III. ESOC Operations Department Organization

The Operations Department at ESOC is organized to provide efficient support to the missions. It is split into a number of divisions, including system development, ground stations, communications, and mission operations support. For each mission these divisions have to work together to provide all of the elements required to support the launch and operations of an ESA satellite. In addition the divisions, especially mission operations, have to interact with the specific project departments at European Space Research and Technology Centre (ESTEC) and European Space Research Institute (ESRIN) to ensure that all interfaces and data transfer mechanisms, for satellite and data processing, function harmoniously.

A. Previous Organization

Until recently, for each mission a Ground Segment Manager was appointed, with a dedicated team and dedicated facilities for each mission. Although systems and lessons learned could be inherited from previous missions, it was not always guaranteed.

B. New Organization

One manager is responsible for the FOS development and preparation for all EO missions. Working for the Head of Division is a team of Spacecraft Operations Managers (SOM) who are responsible for the day-to-day management of the FOS development and implementation. A pool of engineering staff, the Flight Control Team (FCT), works for the Division Head, who may be allocated to work on a specific project for one SOM or who may be used as a general resource for several different projects. Reporting directly to the Division Head, but working in conjunction with the SOMs and FCT, is a centralized Project Control for all EO missions and area responsibles nominated for the various elements of the FOS, including mission control systems, simulators, flight dynamics systems, ground stations, and networks.

The SOMs were initially nominated to be responsible for a single project, but with the commencement of new programs they provide support to the Division Head in the definition phases of the next generation of EO missions, for example Swarm, GMES, EarthCare, thereby facilitating the transfer of lessons learned.

IV. Family of Missions Concept

Although the objectives and specific design of the EO missions are different, the FOS elements have been considered from the beginning of the CryoSat and GOCE development phases to have a common concept. The missions may require specific implementations to fulfill some requirements, but the overall operations concept has been applied to all of the new EO missions. These concepts include:

1) Use of generic MCS, based on Satellite Control Operations System 2000 (SCOS-2000) evolution, for the entire EO family;

2) Automated remote control of dedicated EO routine phase ground stations and FOS systems;

3) Sharing of common facilities among the family (network and ground support equipment/dedicated control and support rooms);

4) Reuse as much as possible the external interfaces and exchange mechanisms;

5) Establish a one station support concept (if this supports mission requirements);

6) Use of standard tools for operations preparation in industry and ESOC (procedures and satellite database editor);

7) Standardize all EO ground segment related documentation;

8) Standardize all ground segment reviews;

9) Promote spacecraft compatibility with ESA radio frequency (RF) and Packet Utilization Stands (PUS) standards;

10) Adopt generic FOS and operations concepts, using same preparation methodology; and

11) Share team members across the family.

This concept has been applied to the EO missions, even though they have different objectives, orbital characteristics, satellite design, and payload return requirements. This can be seen in Table 1.

A. Use of Generic Mission Control System

To monitor and control the satellites in orbit, ESOC has developed the Spacecraft Control and Operations System (SCOS), initially SCOS-1, but for recent missions

Table 1 CryoSat, GOCE, and Aeolus mission characteristics

Objective and payload	CryoSat Ice thickness trends radar altimeter	GOCE Earth gravity field and geoid gradiometer & sat-sat tracking instrument	Aeolus NRT wind profiles doppler wind lidar
Altitude [Visible Passes/day // max/av pass (mins)]	717 km [12 // 12/8.7]	~250 km [6 // 6/4.4]	400 km [10 // 7.8/5.5]
Science Data Downlink	X-Band	S-Band	X-Band
Science Data Generation Rate	400 Gbit/day	0.7 Gbit/day	5.2 Gbit/day

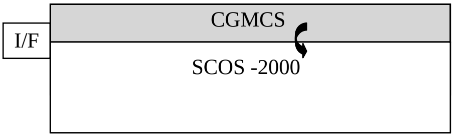


Fig. 1 CryoSat/GOCE mission control system.

SCOS-2000. This is based on a distributed client-server architecture and covers generic services for: 1) telemetry reception and processing, 2) telecommand uplink and verification, 3) data archiving, display, and retrieval, and 4) data distribution and dissemination.

The SCOS-2000 Mission Control System has been used by a number of missions at ESOC, including Integral, Small Missions for Advanced Research in Technology (SMART-1), Rosetta, and Mars Express. SCOS-2000 provides a base for every new Mission Control System (MCS) developed by ESA, on top of which mission specific functionality has been added for each mission. However, improvements that are developed as mission specific may take several generations of SCOS-2000 to be available to the other missions.

It was decided that for the EO missions, a combined MCS system would be developed from the beginning, with requirements from both CryoSat and GOCE used in the initial definition. This resulted in a two-tier approach using the SCOS-2000 kernel plus CryoSat and GOCE specific functionality (Fig. 1).

Improvements coming from the CryoSat and GOCE Mission Control System (CGMCS) development were repatriated back into the SCOS-2000 kernel. When development for the next EO mission, Aeolus, began, it was decided to expand the concept further to make it generically applicable to all future EO missions. This resulted in a three-tier system for the mission control system (Fig. 2).

When new elements are developed for a specific mission, they can later be introduced back either into the Earth Explorer kernel (d) or to other mission specific elements as applicable.

Elements that are incorporated into the EE kernel can also be introduced back into the SCOS-2000 kernel if they are deemed useful for the other missions at ESOC (e).

This allows following Earth Observation missions (e.g., Swarm, Sentinel, EarthCare) to take immediate advantage of the development performed and tested

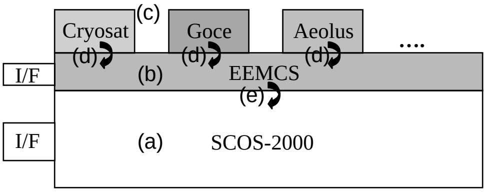


Fig. 2 Earth Explorer mission control system.

by mature missions (e.g., CryoSat, GOCE, Aeolus), but does not restrict them if they require further functionality in order to fulfill specific objectives or take advantage of newly implemented PUS services. Additionally, it is possible to disable any functionality in the Earth Explorer Mission Control System (EEMCS) kernel, which is not applicable for a specific mission control system.

From a cost perspective, this approach generally decreases the cost of successive mission control systems, since a large part of the basic functionality exists and so the new development is limited to specific requirements for the mission or the integration of new functionality from a later version of the SCOS-2000 kernel. This can be seen in Fig. 3, showing the relative costs of the CryoSat, GOCE, and Aeolus Mission Control Systems [1].

It can be seen that both the GOCE and Aeolus mission control systems cost less that the initial CryoSat specific system. However, the Aeolus MCS costs slightly more than the GOCE specific MCS, but the Aeolus overall mission is more complex and required a significant number of mission specific functionalities supporting new PUS services. In addition, since approximately 70% of the base requirements were already part of the SCOS-2000 and Earth Explorer MCS kernels (parts a and b in Fig. 2), improvements in the man-machine interface have been included to enhance the analysis and monitoring functionality. With the new functionality and interface improvements available as part of the EEMCS kernel in the next generation, it is likely that the relative mission control system cost for the next EO missions (Swarm, Sentinel, EarthCare) will reduce further.

B. Automated Remote Control of Ground Station Equipment

One of the main cost-saving concepts for the EO missions is the reduction of manned coverage to working hours. For CryoSat, it was planned to have only Spacecraft Controller (SPACON) coverage and engineering staff during normal working hours. However, the passes outside of this period would still be taken automatically by the control systems and ground station facilities.

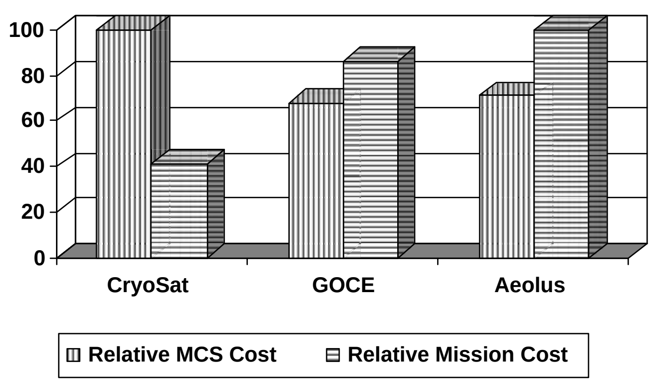


Fig. 3 Relative Earth Explorer mission control systems.

To facilitate this concept, the FOS ground segment, including MCS and station scheduling, as well as the Payload Planning system, had to be designed with this in mind (see Fig. 4).

The inputs for the mission planning are received from the Payload Data Segment (all payload operations), Flight Dynamics (all orbit information and events), and the Ground Station (Station Unavailability Plan). These files are merged within the Mission Planning system, which is a part of the mission control system. From this, three output files are created: Execution Based commands (On Board Timetag), Release Based Autostack commands (On Ground Timetag), and Station Schedule. A copy of the overall timeline is sent back to the Payload Data Segment to verify that the Mission Planning Schedule is in line with the expectations, but also as the baseline against which the monitoring and performance analysis of the instruments is performed.

The Execution Based commands are the classic timetagged payload queue Telecommands (TCs), uplinked by the SPACON in a large burst to the onboard mission timeline, to be executed by the satellite at the specified time. Many hundreds of commands can be released during one pass, reducing the number of visibilities that require active SPACON involvement or a Telecommand uplink.

The Station Schedule File contains all of the timing events to configure and control the station equipment, for example, pre-pass testing, link connection, antenna pointing angles, and acquisition times.

The Release Based Autostack commands contain the TCs that would normally be sent by the SPACON once the satellite is in visibility and a TC uplink from the

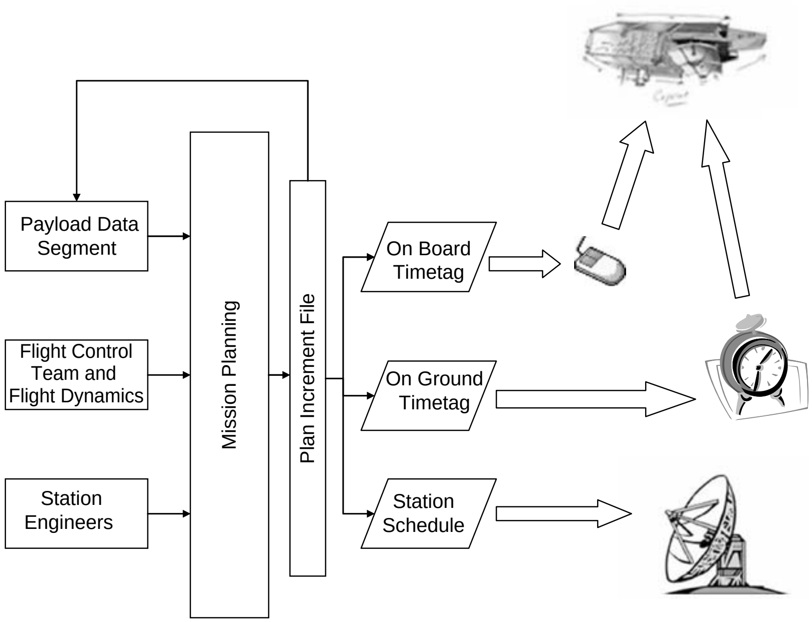


Fig. 4 Mission planning and automation.

ground station has commenced. However, for CryoSat these commands wait in a special queue on the MCS until the planned execution time has elapsed when they are released from the Autostack as immediate commands. In addition, pre-transmission validation is performed to ensure that the satellite is in the correct configuration to receive the commands, preventing inappropriate TCs from being released if the satellite has suffered from an anomaly. Using this functionality, the Mass Memory, which contains all of the payload and platform data that has been stored while the satellite was out of visibility, can be dumped to the ground for automatic transmission and processing at the Payload Data Facility and the FOS, respectively.

During working hours the SPACON is available to verify the correct execution of the various operations in real time, but because the majority of the tasks are handled automatically, the Spacecraft Controllers can be employed to perform higher level data collection and analysis functions, which then releases the engineers from routine and time-consuming activities.

C. Standardized Tools

One of the more time-consuming aspects of the operations preparation for launch is the translation or generation of the satellite database and operations procedures. As part of the CryoSat contract it was agreed with the prime contractor (Astrium) that common tools would be used for both of these elements.

The SCOS-2000 database editor was provided to Astrium, and so the Assembly, Integration, and Test (AIT) and Operational satellite databases were generated from a single master reference database that contained an agreed set of tables and fields. This meant that the operations database provided to ESOC was directly compatible with the SCOS-2000 systems so that no further translation was required, reducing the possibility of errors during import. In addition, because this was the same as the database that had been used during the satellite testing by the AIT team, it had been validated on the spacecraft prior to delivery to ESOC. Although the Flight Control Team performed their own verification and validation of the database, it was of a consistently high quality, and so the majority of work was to introduce “value added” elements, for example, derived parameters, synoptic display definition, command sequences, and sophisticated limit and pre- and post- transmission validation checks. Information regarding any small errors that were found could be passed back to Industry, so that the AIT could also benefit from using the same database.

In many projects the operational procedures are included as a textual part of the Flight Operations Manual (FOM), which the FCT then have to write in the procedure tool for that mission. For CryoSat it was agreed that the Prime Contractor would provide a set of operations procedures, including nominal and contingency recovery procedures, using the same tool that the FCT was using, namely Mission Operations Information System (MOIS). These procedures were validated by Astrium using different test tools, including the Satellite Validation Facility (a software validation environment), the Real-time Test Bed (a mixture of hardware and software), and the satellite itself. Again, this led to a high quality of procedures being delivered to ESOC in the correct format and allowed the FCT to spend the time to tailor and enhance them for the EO operations concept rather than starting from a textual input [2].

D. Standardized Documentation and Reviews

The preparation of documentation to support the definition and development of the systems for the satellite operations can be very time consuming. Even though the various projects of the EO Missions Division have different objectives, requirements, and satellite designs, by keeping the supporting documentation consistent, time can be saved by ensuring that all necessary elements are covered in sufficient detail to guarantee success. If each project determined its own format and content of the documents for the preparation activities (including requirements definition, architectural design, test plans, and reports), it would be less easy to gain the benefit from previous missions and also using the previous team members to assist in reviews. Of course, lessons are learned from each mission so that the documents are improved and enhanced, but by maintaining a consistent approach time and effort can be saved.

At various stages of the Ground Segment development there are critical reviews: 1) Ground Segment Requirements Review (GSRQR), verifying the requirements definition and consolidation; 2) Ground Segment Design Review (GSDR), verifying the design and development approach; 3) Ground Segment Readiness Review (GSRR), verifying the successful test and validation of all elements; and 4) Operational Readiness, final verification of all systems prior to launch.

By standardizing the reviews, including objectives, content, and format, the different teams know in advance what is expected and can prepare accordingly. Because the documentation is standardized, the contents of the datapackage, which is prepared as part of the reviews, is consistent, thus making it easier for external reviewers to determine the state of preparedness of the Ground Segment at various points in the design, development, implementation, and testing phase. In this way, problems are less likely to be masked, because there is already a yardstick against which the state of the mission can be measured, i.e., the state of preparation of a previous mission at the same stage.

E. Team Sharing

As previously mentioned, there is a pool of engineers working in the Earth Observation Division Flight Control Team at ESOC. Members of the team may be initially dedicated to one mission or may provide a specific function to several projects simultaneously.

The CryoSat team experienced the complete set of preparation activities (definition, implementation, and test) for all phases up to the launch. Additionally, the GOCE team members were involved in the launch preparations of CryoSat as the secondary team, so that they have also experienced the full gamut of activities for the preparation of a mission. Thus, there is a significant pool of well-experienced engineers in the Earth Observation Division. They can be used in the review and definition of new projects, ensuring that the requirements and proposed design will be appropriate for the mission, knowing the possible problems and potential pitfalls in advance, since they have been through the complete process already. An added bonus is that they already know the tools and concepts used in the Earth Observation Missions Division, and so the initial learning curve is limited to the specifics of each mission.

Within the Family of Missions concept is the planned transfer of personnel from a flying, routine mission, to a project that is in the early stages of development. This provides not only continuity of personnel for the flying missions, but also additional work interest, since the engineers are not limited to only performing routine operations. In the event of anomalies on the flying missions, the original, experienced staff are still available, while if the mission is very tranquil, the engineers can concentrate on the next project, possibly improving efficiency for the flying and future missions by implementing new automation systems.

Following the CryoSat launch failure, it was possible to absorb the team directly into the GOCE and Aeolus FCTs with very little impact on the overall cost of the missions. This was because the transfer had already been planned; it was just a little earlier than expected.

V. Conclusion

Although the specific objectives and satellite design of the ESA Earth Observation Missions may vary, by defining a harmonized concept that is applied to all new missions at the definition and requirements phase, the cost of the ESOC support elements can be reduced. The ability to share personnel and equipment during periods of high activity allows for increased flexibility and reduced infrastructure requirements. Additionally, once the team has experienced one complete life cycle of a project, the potential pitfalls and problems can be avoided or minimized during the design and implementation of the following projects, thus reducing resources required to solve last minute or unexpected problems. While the Family of Missions concept is not unique to ESOC, it has been demonstrated within the frame of the Earth Observation Missions Division that it is a very successful way to manage personnel and resources for operations at ESA.

Acknowledgments

The authors wish to acknowledge Klaus-Dieter Mau and Bernd Koesters, Astrium, Germany, and Kate Adamson, Science Systems, Germany.

References

[1] Reggestad, Guerrucci, "Earth Explorer Family Mission Control System: A New Child Is Born (Aeolus)!," AIAA Paper 2006-5682, June 2006.

[2] Cater, J., "CryoSat Mission Preparation: A Cost Savings and Quality Improvement Case Study," AIAA Paper 2006-5922, June 2006.

This page intentionally left blank

Chapter 2

Maximizing Commercial Space Transportation with a Public–Private Partnership

Thomas C. Taylor*
Lunar Transportation Systems, Inc., Las Cruces, New Mexico

I. Introduction

THE transportation market from Earth to and from low Earth orbit has evolved to the emergence of commercial entrepreneurs hoping to offer transportation service to the general public. Most people expect low space transportation costs will come quickly, but the commercial markets and entrepreneurs bring a multitude of changes to the status quo with the lowering of costs as a later visible result and not the quick solution to the high cost of transportation to orbit. Public–private partnerships (PPPs) may offer commercial space entrepreneurs a place at the procurement table. Some of the items entrepreneurs bring include innovation in hardware, and a maturing of the normal commercial market forces such as the pressures from buyers and sellers rather than those from government planners or from regulation.

It has been 60 years since 6000 V-2 rockets were produced in World War II, and no other rocket has achieved that number. The upgrading of expendable rockets has given way to partly reusable space shuttles and possibly fully reusable launch vehicles on the commercial development horizon. Launch costs are still high after 60 years, society wants orbital hotels, and current/future markets are not emerging because of high transportation costs. NASA proposes a new Ares transportation vehicle approach, and stimulates commercial launch vehicles to transition to newer, more affordable launch innovation with private financing and sustainable commercial Earth-to-orbit (ETO) development, so that NASA can proceed to the next goal beyond. Entrepreneurs need stronger agreements with government to raise the private money required for commercial projects in the future.

The first space transportation cycle to low Earth orbit (LEO) is the toughest due to the gravity well. Expendable evolved launch vehicles (EELVs), NASA retiring

*Vice President. Senior Member AIAA.

hardware, and/or the emerging NASA Ares vehicles provide NASA and commercial entrepreneurs with a public–private partnership opportunity to stimulate several types of transportation commerce innovation starting with real cost reduction innovation to produce the lower cost space transportation everyone desires. The public–private partnership concept is proposed with each side bringing "skin in the game" and deriving benefit through a variety of concepts. One concept, a general type immediately available, is a public–private partnership for modification of the space shuttle and other space commerce projects to accomplish a commercial transition strategy. This first example PPP concept proposes an unmanned cargo carrier with the commercial payloads similar to the two outside vehicles in Fig. 1 converted via a commercial transition strategy (CTS) within a PPP to launch by the commercial financial organization, which also uses the NASA-KSC staff and contractors to ease the human transition to Ares I after 2010.

The government budgets around the world limit these governments to one or two innovative launch solutions financed by governments, when dozens of transportation concepts financed by private money should be stimulated commercially with the same government money and can be in the PPP sector using these methods. PPPs worked in the New Zealand government and could work for space commerce development. Applied to aerospace operations and commercial business, PPPs provides a partnership that pays taxes rather than requires tax budgets. Take, for example, a commercial transition strategy approach for space hardware similar to the military transition strategy that works so effectively for the military organizations around the world, where eventually space hardware becomes space or war surplus hardware available to commercial and private market sectors for a

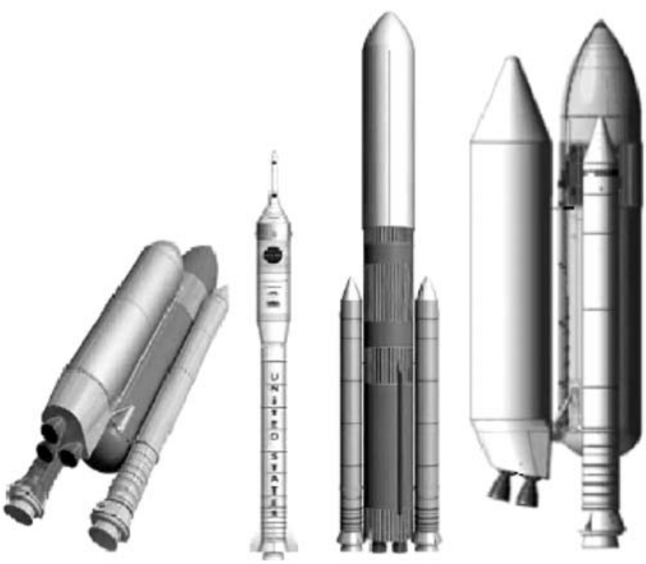


Fig. 1 NASA Ares I and Ares V in the center with the proposed Shuttle Cm on either side.

price. (The commercial transition strategy is a term suggested by the author and probably not a term seen elsewhere.) In the space hardware business, governments place transportation systems like the space shuttle on the scrap heap rather than making this hardware available in some way for use by commercial and private organizations and individuals. It is hard for governments to cut the labor costs, especially when they are doing jobs that are more effectively done by non-government forces. For example, commercial space companies like SPACEHAB, INC., have successfully reduced the cost of "manned tended research in the shuttle" by an order of magnitude with commercial operations by introducing commercial staff and innovation. SPACEHAB's hardware was 9 times less expensive to build and operated 10 times less expensively.

Lunar Transportation Systems, Inc., is a concept of commercial payloads launched on commercial vehicles to LEO and traveling to and from the moon's surface. The future acceleration of space exploration commerce is difficult to predict with regard to the innovative use of public–private partnerships, but New Zealand [1, 2], for example, has made history in turning their Forest Agency and country around using methods of combining government markets with commercial operations and funding. Propellant depots are proposed as possible PPP opportunities.

Public–private partnerships [3] change the way taxes are created and spent, but the real value is the cooperation between private industry and government in the larger markets created, and the flow of private money into the areas of cooperation now requiring 100% tax based budgets to satisfy the government space requirements, establishing a more commercial operation and the stimulation of the follow-on commercial markets. Internationally, PPPs can also work commercially.

II. Public–Private Partnerships (PPP)

Public–private partnerships [3] offer an "in between" opportunity for NASA, global space agencies, and all governments to transition from the government space budget world of paying the entire cost of the development of new space transportation hardware to just buying space transportation services from private sources allowing government budgets to be used for different future activities. Buying the commercial services proposed rather than paying 100% for all hardware up-front with government budgets means government budgets stretch further. Thirty years ago such commercial transportation services were not possible, because no real commercial space services were available. Now commercial entrepreneurs like Burt Rutan, Richard Branson [4], and others are proposing that such services are potentially available from the commercial sector. While entrepreneurs usually have their own innovation and pricing, these private sector innovators bring private money and innovation to each space market.

Public–private partnerships are emerging in various forms around the world including wide use of PPPs in municipal projects. One general type of public–private partnerships [3] allows each participant to bring something to the table and each participant derives some benefit from the cooperation. The items brought to the table can vary widely, and each PPP is a somewhat unique agreement between the public and private participants, provide an opportunity for technology sharing and significant cost reduction, plus offer new opportunities for entrepreneurs and

governments. The financial opportunity is to leverage the government participation with the private money and get mankind into the rest of the universe using projects too large for individual organizations on either side of the public-private boundary.

The public–private partnership provides an opportunity for a commercial transition strategy or plan to gradually move from government to commercial operations with government and commercial customers sharing the cost of commercial service overhead. These partnerships offer the opportunity for many smaller companies to develop many innovative technologies leading to services like space transportation, providing the government with the ability to pick the services that show promise, and by using the government budgets to define space launch markets for private capital investment rather than picking one specific concept and spending all the government budget on a single attempt to create the next space launch vehicle or technology concept. These global public–private partnerships may provide the opportunity to leverage the government budget participation, stimulate many innovative technology developments, rather than just one alternative, use government funds to pay for services actually reducing costs, and focusing government budgets for items not capable of being financed with private funds.

A. Commercial Transition Strategy (CTS)

The Commercial Transition Strategy is an agreed-upon plan by interested parties, organizations and governments for transitioning from the current government hardware from government ownership and operation to the commercial sector for the purposes of realizing additional value from the technology and the hardware originally funded by the taxpayer and now ready for retirement. The DOD performs this transition every day by selling used military hardware and other items as “war surplus” and/or offering the same to allies and other countries in return for some value. It is hoped that NASA program hardware would be of value after the 100-mission design life of the hardware is reached, and a commercial organization could make the transition to profitable commercial operations earning revenue and generating taxes. It should be noted that the National Space Transportation System (NSTS) design also had a time requirement, like 10–20 years after which the hardware was deemed to be too old regardless of the 100-mission design life. The shuttle orbiters are at approximately one-third of their 100 missions. Some technology used in the shuttle could be upgraded given the advances in commercial microelectronics, but the recertification costs are prohibitive. The key to CTS is early planning before the hardware design is final, so that the “second use” concept can be made appealing to private money investment and the production of hardware can be accelerated, which may be a little like selling fighter jets to foreign allies to keep the production lines operating longer. The Hubble Telescope [5] might have become a follow-on commercial operation, if the commercial transition strategy had been drafted early by the Hubble [5] operators or some other organization, and these commercial people were given sufficient time to plan and raise private funds for follow-on commercial “for profit” operations of the telescope and the generation of tax revenue. Skylab [6], the space shuttle, and the International Space Station (ISS) are all examples of potential opportunities for commercial transition strategies that may have slipped by our country.

What can public–private partnerships do and why use them? PPPs stimulate the private financing entry into space commerce by adding stability, larger markets, and investor comfort in space commerce. Private financing can bring commercial innovation, cost reduction, and international markets within the right PPP framework. This chapter proposes several PPP examples possible in the future created from known commercial organizations interested in commercial operations. These interested commercial organizations propose the PPP scenarios discussed, and the concepts include a side-mount Shuttle Cm, a habitation testbed to test future hotel habitation services, entertainment, lunar transportation logistics and surface structural/mining/processing systems, a nearer term propellant depot with an interested first and second commercial customer, tourist facilities, lunar in situ resource utilization (ISRU) and space solar power futures options, and others.

B. Global Space Projects for Global Benefit

Space projects have become too large and expensive for individual organizations. A number of global space projects like commercial propellant depots, LEO tourist habitation, space solar power (SSP) [both geostationary Earth orbit (GEO) Earth and lunar resource assisted], commercial lunar development, and others are technically feasible, but not a goal within NASA or even other space agencies or the government budgets around the world, and are not financially feasible due to the high costs of ETO transportation, aerospace manufacturing, national goals, and government procurement. Globally, we are experiencing, in general, an ever-increasing rise in the cost of space programs, and this high cost limits space activities at a time when mankind should be exploring and developing options off our single planet. Commerce and private money can be introduced and has been successful in ventures like ComSats, where one launch provides almost instant revenue flow and profits. The future projects just mentioned require more than one launch, greater risks, and increased costs. Seventy percent of a lunar mission is getting out of Earth's gravity well. Examples of applying the public–private partnership model [3] to stimulate lower costs in both ETO transportation and beyond are summarized. Each commercial concept could be the result of an innovative public–private partnership solving global energy, technology, and transportation problems, which can be accomplished in cooperation with government. Some projects could be innovative international public–private partnerships used to stimulate space solar power projects of a global nature with international financing and far reaching benefits for mankind and the atmosphere we all share.

Various example PPP projects could include a near-term commercial transition from the space shuttle to an unmanned cargo carrier version called Shuttle Cm [7], commercial propellant depots [8] in LEO, space solar power [9] fabrication/assembly, and development of a commercially sustainable lunar transportation system [10] focused on long-term transportation use and phased cost reduction plus the recovery and development of lunar resources. Some of these projects are detailed with graphics showing the PPP structure in [11]. A public–private partnership, simply described and defined here, is proposed by entrepreneurs, and combines the resources and markets of government and private sector to bring private investment and business practices to benefit both the public and private participants.

When services are contracted out to private companies, doesn't that mean that public employees lose jobs? The answer is no, the Department of Labor examined in a 2001 report and found that public workers don't lose jobs because of public-private partnerships. Examining partnerships in 34 cities and counties, the Labor Department found that virtually all affected public employees were either hired by private contractors or transferred to other government positions. In fact, the most productive partnerships have been those in which government employees (and sometimes their unions) are actively involved in the partnership planning process. New Zealand's methods appear to work well elsewhere [1, 2].

Space tourism is even becoming an issue. After the wealthy pay a fortune to ride to 100 kms, space tourism vehicles will be in production to transport many people on a 30-minute ride to orbital hotels capable of making the orbital destination worth the cost. For society it is six days and seven nights as the entire world turns under their own stateroom. The trips will become more affordable as the market space transportation matures.

A public–private partnership can become the basis of cooperation between governments and commercial forces in projects and industries deriving benefits from space. The transition from governments paying all the development cost to sharing the burden with private investors can be viewed as a commercial transition plan using public–private partnerships to address many issues sometimes difficult for both governments and private industry.

C. New Zealand Example

Tax reform [1, 2] in New Zealand might provide a good model for the governments to study for space and possibly to provide a direction for change in tax reform [12] and embracing the public sector. At the beginning of the 1980s, New Zealand had a system of high tax rates and lots of loopholes within their tax system. This led to high taxes, economic stagnation, high unemployment, and fiscal disarray. In the 1980s the New Zealand Labour Government adopted a Reaganesque formula for tax reform with the top personal income tax rate being reduced from 66% to 33%. The tax base was extended and most loopholes removed. The corporate tax rate was also lowered from 45% to 33%. The alignment of the corporate and the top rate of personal tax, along with the integration of dividend income through the imputation system, produced a system with no double taxation of corporate income.

The results of economic liberalization were striking. Along with strong economic growth, unemployment dropped and budget deficits turned into surpluses. The results were so dramatic that other governments started visiting New Zealand and are exploring similar systems. From a layman's view, here is how it was done in one segment of New Zealand's government. The New Zealand government, for example, had 5000 people in their government Foresty Agency managing their forest assets. The government tax rate was in the 30% average person range with the wealthy paying over 60%. The New Zealand government reduced the forest staff on the government side to five people, while procuring services from the private sector. This produced commercial jobs for the government forestry people and worked so well in bringing down the cost of a portion of government that it was tried in other sectors of their government where commercial organizations could help. The country's tax rate is now 11% and other countries come to New

Zealand to see how the same strategy can be applied in their countries. There are things that only governments can do effectively, like fighting wars and regulation, but the tendency is for governments to grow in size and to do tasks that private industry can do more effectively, given the market forces of the commercial environment. When governments do jobs capable of being performed by commercial forces, the government requires more tax budgets rather than commercial companies profiting and paying taxes supporting the government budgets. It is not that simple, but in the space launch vehicle industry, where commercial organizations have become more capable, the public–private partnerships can help even without other changes in tax reform. Let's face it, the American taxpayer has supported space development budgets since World War II, but may one day realize the return on that investment, on 60 years of investment, has not created the commercial industries the same way the aviation or computer industries have.

III. Shuttle Cm Proposed PPP

The space shuttle hardware system, called the NSTS, consists of three primary components: the reusable orbiter, the discarded external tank, and the twin solid rocket boosters, expended with the casing recovered from the ocean, transported to Utah, cleaned, filled with propellant, and returned to Florida. NASA Kennedy Space Center facilities and a large staff of NASA and contractor employees are an important part of the government NSTS operation that generally require pay regardless of the launches scheduled and the support required by these launch hardware components. Proposed here is a PPP sharing of overhead of all kinds and the PPP exploration of what else could lead to reduced cost and increase efficiency. Not everybody within the status quo would see this PPP sharing as a positive step, but the right PPP could have a significant impact on commercial costs and the costs currently borne by government.

A. Creating a Commercial Transition Strategy for the NSTS

A commercial transition plan takes government hardware with some remaining value, but being scrapped, and transitions it to the private sector in a manner allowing some additional value to be realized by a commercial venture and the government. The military successfully transitions hardware for reuse, why not space transportation hardware by NASA? The DOD is effective in transferring hardware of many types to the National Guard forces, allies, and even as war surplus. New military hardware is sold to foreign allies with ease and little International Treaty in Arms Reduction (ITAR) complications. Exploring the transferring of NSTS hardware already built and on the scrap heap to the private sector to get value from the remaining design life is proposed. In both the civilian and military hardware transition strategy to commercial operations or wars surplus sales, the civilian customer or second user is so far into the future that this final customer is never consulted about design requirements, design changes to make the commercial transition easier or increase the value of the hardware to the second user, or changes that might increase the market value for the hardware. What happens when a commercial space entrepreneur is asked for commercial modification suggestions after the space launch hardware has had an active life within government service?

B. National Space Transportation System External Tank PPP Proposed

Figure 2, for example, begins a look at the existing space shuttle components at a point just before the scrap heap with little chance for an active transition into commercial use and few if any private sector organizations willing to step in and save the NSTS from the scrap heap. The current external tank is discarded in space to reenter and 80% burns up upon reentry with the remainder impacting the ocean surface disposal area northwest of Hawaii. Focusing on the Shuttle Cm ET component, Fig. 2 answers the questions of what a commercial space entrepreneur would do with the external tank as part of a $12 billion vehicle (probably 1980 dollars) with over half of its original design life remaining. Figure 2 starts the process of a commercial transition strategy using the external tank as one element and provides an external tank PPP evolution path of a government space launch vehicle to a commercial market, where a private money organization and future customers come forward and provide shuttle derived vehicle upgrades of a commercial market nature to satisfy a future market for the commercial unmanned Shuttle Derived Vehicle (SDV). This figure evolves only the external tank (ET) by converting the ~58,000 pounds of discarded tank, 82% aluminum, into a salvageable, salable item on orbit, evolving from one payload bay volume (PBV) now to a future of seven

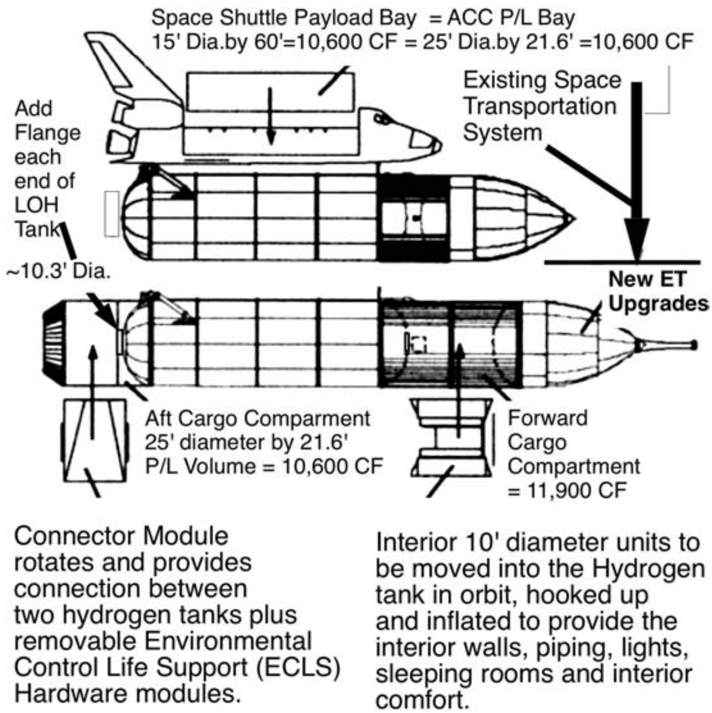


Fig. 2 External tank upgrades to an unmanned carrier suggested by an entrepreneur for commercial markets in orbit.

PBVs and possibly altering drag and reducing ice problems with some altering of the aerodynamic flow around the vehicle plus other innovation. Figure 3 evolves the orbiter and proposes an unmanned carrier. The flow is changed around the vehicle using the aerodynamic spike and fluid aerospike (aerospike, an unfortunate choice of words by the author and used in the original Air Force SBIR work at the Air Force Academy Tri-Sonic Wind Tunnel and many times confused with the aerospike of a jet engine). Some drag reduction may increase the future payload weight capability, but the real advantage may be the seven-fold increase possible in payload bay like volumes and the move to 25-ft-diameter payloads possible in the assembly of space solar power units in LEO, which are low-weight, high-volume payloads with the commercial launch repetition to sustain a commercial transportation service and the private investment required.

It should be noted that a two-year industry study funded by major aerospace organizations involved with NSTS support did suggest a "Shuttle C," side-mounted concept," but their "Shuttle C" did not radically force commercial innovation and commercial evolution, and did not result in greatly increased payload volumes or increased commercial markets. This industry study was very good technically, but was unable to change the NASA system or reduce space transportation costs or increase commercial markets for an existing vehicle or convince NASA not to scrap the ET rather than to evolve into using private sector help rather than relying on 100% government budgets. This industry study was a little like Abraham Lincoln in the 1860s asking the existing American railroad companies in the East to extend their tracks to the Pacific and getting the answer back, "Good idea Mr. Lincoln, but we need money for a bridge across the Missouri River at Omaha."

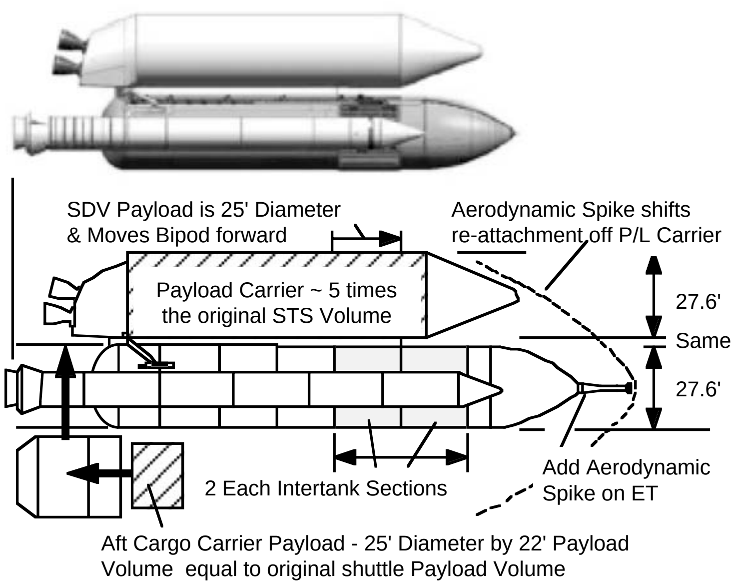


Fig. 3 Orbit upgrades to an unmanned carrier for commercial markets in orbit.

Rather than paying 100% of the bridge and the costs of a Transcontinental Railroad design and construction, the government, while fighting an expensive civil war and fierce internal national politics, took a different approach, not totally unlike a PPP. The government and President Lincoln stimulated hungry commercial entrepreneurs, created a railroad race, gave future wealth in form of land and recoverable resources at numerous milestones, and got their railroad from the private money contractors of the day, who sold land before they got it, created towns over night, opened the nation's future options for growth, and got their Transcontinental Railroad before the bridge at Omaha, suggested by the established entrenched railroad industry, was built. President Lincoln leaped beyond the existing railroad industry, started on the west side of the Missouri River, created a two-company race starting one non-railroad company at each end of the longest railroad built in the world to date, worked out PPP-like financing and started it all during the costliest war in United States history. A book by Stephen E. Ambrose titled *Nothing Like It in This World*, about "The Men Who Built the Transcontinental Railroad 1863–1869," details the epic struggle that eventually built 10 miles of track in a single day [13]. The resulting trade route across the nation reduced transportation costs, stimulated development of the Louisiana Purchase, transported in both directions with immigrants/settlers west and cattle east, received government stimulation from items the nation could easily give at the time, recovered natural resources, changed our nation forever, and helped propel our country into a leader nation. Now a very similar situation exists, as mankind prepares to move off the planet in a sustainable manner, and we need the same courage in commerce and government innovation in stimulating the commercial entrepreneurs of America to assist in the space exploration initiative.

The early ET market in orbit is focused on ET attributes leading to a series of possible uses capable of evolving in many different directions. Some of these attributes are:

1) ET payload diameter: 25 ft payload diameter by 22 ft long as an option via the ACC;
2) ET FPR propellant: recover 25k to 50k flight propellant reserve for ISS or other commerce;
3) ET GG stability: uses the 154 ft ET length to provide long axis gravity gradient stability;
4) ET volume: uses the interior 27 ft diam by 100 ft long volume for various orbital uses;
5) ET base metal: uses the 82% aluminum in the ET as base metal source via melting, etc.;
6) ET hide: uses the hard vacuum shielded interior to obscure objects from view;
7) ET mass: uses the 58,000 lb as mass for other operations;
8) ET truss: uses the ET as a portion of a truss member given its hard points;
9) ET economy: uses the $580 million of invested transportation energy already paid to eliminate other launches.

The ideas for exploiting of the external tank in orbit can take many forms. See Fig. 8 of [11] for an expansion of each of the preceding ideas. Taking one attribute like internal volume and expanding into it results in the following concepts and markets.

Basically, commercial customers for the salvaged ET appear excited and able to visualize final uses, but are unable to break through all the barriers related to the ET basic unit in orbit either unmodified or modified. It should be pointed out that changing the ET before launch triggers a recertification of the ET within the NASA environment, which in the past has cost over $1 billion and could cost double or more than that dollar amount now. The recertification cost in the commercial environment is not known and depends on the changes. The potential value of the commercial salvage of the ET basically centers on what existing/ future attributes of the ET are wanted combined with the financial feasibility enhancement of the $580 million in transportation costs already paid. Few in traditional aerospace accept the financial enhancement as a valid reason for the salvage of the ET, but $500 million is significant savings in transportation value to those in the business and financial community.

One attribute of ETs salvaged in orbit is the ET internal volume, which includes the interior 27 ft diameter by 100 ft long hydrogen tank volume of 53,000 cubic feet of volume tested to 40 psi and available for various orbital uses including habitation like space hotels. The public–private partnership attempts to open the orbital accommodation market by proposing the habitation testbed shown in Fig. 4. This testbed introduces the process of exploring the commercial use of the STS components, the ET, the orbiter or an unmanned carrier and the solid rocket boosters.

Expanding that volume attribute with other customers uncovers additional markets. Orbital ETs are used for facilities like the hotels envisioned by the Space Island Group [14, 15, 16] in America. The Fig. 5 hotel uses 30 salvaged ETs with a transportation savings of $15 billion. Future large space hotels will be expensive and currently transportation is a major part of that expense. The Shimizu Foundation [17, 15, 16] in Japan envisions a space hotel in Fig. 6 with even more salvaged ETs. The GLOBAL OUTPOST, Inc. [18, 15, 16] Hotel in orbit can use eight ETs as a torus rotating at 2 rpm to provide near lunar gravity and uses another eight ETs in the axis shown in Fig. 7.

C. NSTS Orbiter PPP Proposed

The orbiter component of the Shuttle Cm is also an opportunity for commercial transition strategy leading to a PPP designed to create a shuttle-derived vehicle (SDV) by evolving from the current space shuttle orbiter to an unmanned carrier and the use of an orbiter public–private partnership to make the transition easier and more productive for both the public, the existing aerospace industry, and the private side of the partnership. The goal is to accelerate new technology like the Ares vehicles to move forward and allow commercial organizations to build on and transition current shuttle orbiter technology to create a more commercial alternative to attractive organizations with private money. Shuttle Cm is but one of many options available.

Figure 3 evolves the orbiter and proposes an unmanned carrier. The flow is changed around the entire vehicle using the aerodynamic spike, and a fluid aerospike helps the ice and decreases the carrier weight below the original orbiter weight to increase the payload weight carrying capacity of the evolved NSTS configuration. The existing 235,000-lb orbiter is removed from the current configuration in phases and substituted with an unmanned payload carrier of less weight

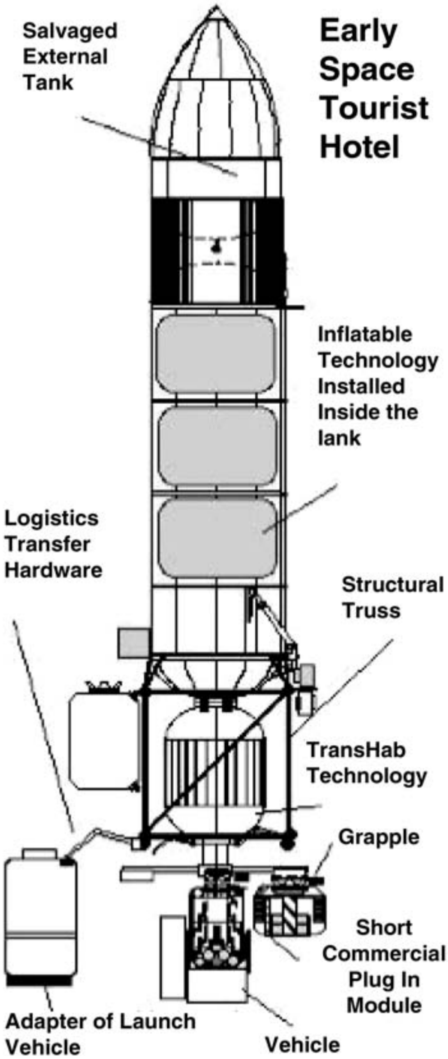


Fig. 4 Hotel testbed from ET to explore costs, logistics, orbital modules, and entertainment.

in the 2010 to 2012 timeframe. The crewed vehicle requirements and crew transport function to LEO is assumed by the NASA Ares vehicles and later possibly by emerging commercial space transportation vehicles. The orbiter PPP contains both public stakeholders like NASA Headquarters, DOD, KSC, MSFC, JSC, Ames, and others and private stakeholders like GLOBAL OUTPOST, Inc., Lunar Transportation Systems, Inc., SPACEHAB, Inc., financial partners, future customers, international customers, and others in a public–private partnership

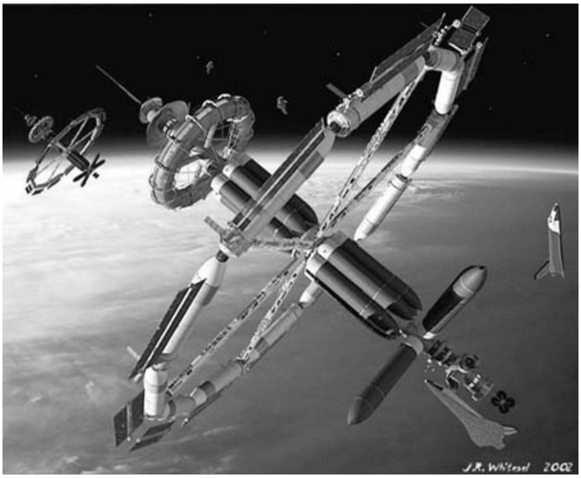


Fig. 5 Space island hotel using 30 ETs salvaged in orbit saving $15 billion in transportation costs [14].

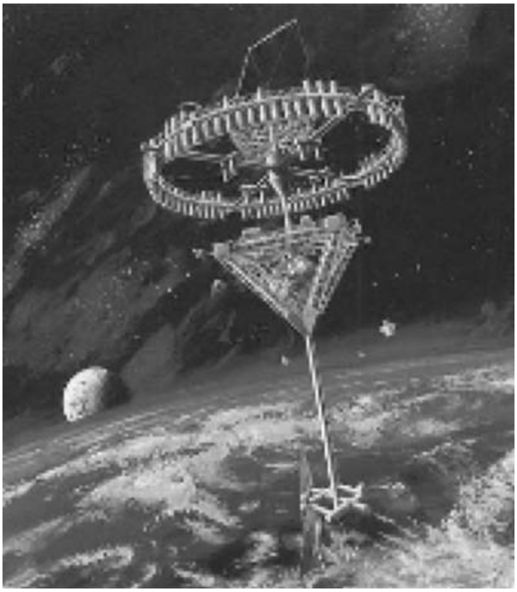


Fig. 6 Shimizu Corporation rendering in 1989 of proposed hotel using ETs salvaged in orbit saving transportation costs [17].

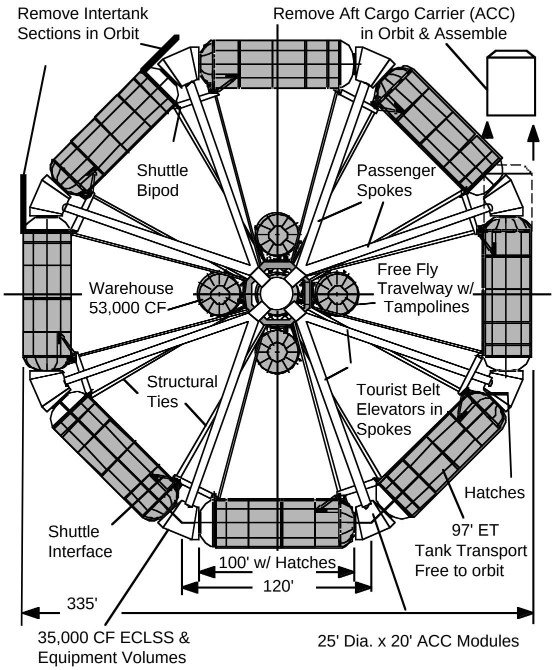


Fig. 7 GLOBAL OUTPOST's proposed orbiting hotel using 18 ETs salvaged in orbit saving ~$9 billion in transportation costs [18].

agreement that evolves toward commercial interest and investment in privately funded orbiter/carrier upgrades. The PPP members usually derive some benefit from and bring something of value to the PPP. The orbiter/carrier upgrades are to improve the commercial market nature of the NASA hardware to stimulate and satisfy a secondary transportation market for the commercial unmanned SDV eventually expanded from one PBV currently to possibly seven PBVs in the future and increasing propulsion with the five segment solid rocket boosters (SRBs), and reduced drag from the aerodynamic spike + fluid spike (aerospike) plus other innovation. The orbiter is removed from the current configuration and operated with an unmanned payload carrier in the 2010 to 2012 timeframe.

Figure 1 (two center configurations) depicts the NASA Ares I Crew Exploration Vehicle (CEV) stick, which is available quickly, and a larger Ares V vehicle available later both using a five segment SRB. Suggested in parallel is a Shuttle Cm (two outside configurations) series of transition steps to a full Shuttle Cm innovation from the private sector to create the sustainable affordable space transportation system and to permit NASA to lead the way to the moon and quickly move beyond to other celestial bodies. This leaves commercial organizations with the ability to develop transportation to and from the lunar surface and to develop the moon's surface resources for commercial use. The later 10-m-diameter payload capability is also attractive as the market grows into larger volume less massive payloads of the space solar power industry.

D. NSTS Solid Rocket Booster PPP Proposed

Creating an SRB PPP is relatively easy by expanding the existing production and transportation operations of the solid rocket booster program and the plans to add a fifth segment to the existing four segment SRB version. The commercial transition strategy for the space shuttle solid rocket booster adds ATK as a PPP member and their operation is expanded and becomes more profitable with the commercial customer using SRBs from the same flow as NASA and both sharing the overhead involved.

E. Results Expected: NSTS Private–Public Partnerships

The shuttle could operate unmanned as an SDV cargo-only transportation business operated without the orbiter by a public–private partnership with the right commercial transition approach and business people with some market cooperation from NASA and DOD and some care not to trample emerging launch hardware innovation or NASA's Ares vehicle plans. This change might give commercial organizations the ability to share the hardware by operating it with NASA as a potential part-time customer for the remaining 10–15 missions and commercial customers able to pay some of the combined operation's overhead costs. This opportunity could be put up to American industry, offering the 10–15 remaining STS missions as a commercial market for the business, approaching Wall Street with a business plan showing the $10 billion pre-existing NASA market, and folding in the United Space Alliance as a potential partner.

IV. Lunar Transportation Public–Private Partnership Example

Additional public–private partnerships on commercial space projects are possible and suggested in less detail than the Shuttle Cm. They include lunar noncritical cargo transportation services, commercial propellant services, and many others.

A. Lunar Transportation System

NASA has conceptualized large Mars sized vehicles for the exploration of the planets. A smaller commercial Earth-moon transportation system may provide the logistics and crew exchange transportation support required after NASA moves forward to other celestial bodies.

Multiple logistics transportation cycles are the lifeblood of remote resource recovery locations on Earth. The North Slope of Alaska, for example, had at least four developed transportation systems, each costing the customer different rates and specialized for specific cargo. For example, humans and emergency cargo traveled at $5/lb on 737 aircraft in good flying weather. Barges around point Barrow and overland trucks up from Fairbanks carried major prefabricated plants at pennies per pound, but also always subject to the whims of Mother Nature. Finally, the pipeline shipped oil at less than pennies per pound. The pipeline transported in 90 days more mass than all other methods combined. The more distant the remote site, the more important the logistics transportation in multiple forms appears to be, including one route that requires minimum planning, reliable

delivery, routine schedules and usually at maximum expense, but emergency human transport, because planners and on-site operations continually have last-minute requirements. [This information is based on personal experience by the author in four years of work (1975–1979) in constructing the Prudhoe Bay Oil Field Processing facilities on the Alaskan North Slope. This work was part of four separate oil company contracts by Peter Kiewit Sons' company of Omaha, Nebraska.]

Lunar Transportation Systems, Inc. (LTS, Inc.), is a commercial transportation company willing to create such an alternative commercial transportation system to the moon, if sufficient public and private components of a partnership can be put in place. A public–private partnership can be established between NASA and other public interests internationally and private investors for a sustainable commercial transportation system [19] affordable for the continuing lunar surface operations including surface resource recovery operations. The LTS, Inc., organization contains the same startup team with successes in starting SPACEHAB, the Kistler Aerospace Corporation, and other commercial space ventures. The LTS startup team includes Walter Kistler, Bob Citron, and Tom Taylor.

This LTS, Inc., system builds the equivalent of a two-way highway between low Earth orbit and the lunar surface. The system uses a small fleet of reusable spacecraft, supported by a small fleet of expendable spacecraft, to transfer payloads in LEO, to transfer cryogenic propellant tanks [20] at relatively stable

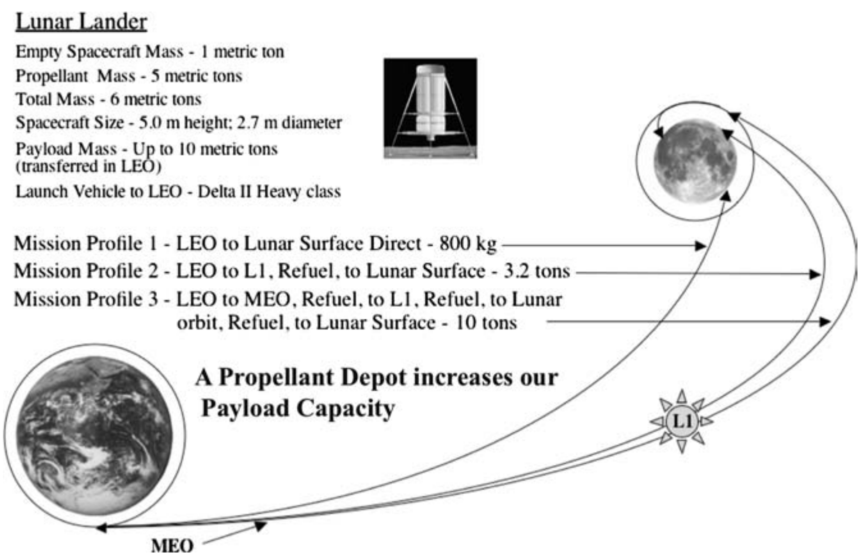


Fig. 8 Lunar Transportation Systems, Inc., proposes a cargo logistics system to and from the lunar surface for nonessential payloads and later sustainable reusable delivery vehicles at commercial rates.

locations in cislunar space, and to transport payloads to and from the lunar surface. The system uses existing ELVs to transport its entire infrastructure including from the Earth to low Earth orbit. Figure 8 describes the basic LTS plan [21].

B. New Lunar Transportation Architecture for Sustainability

Most of the concepts for lunar transportation architecture that are being considered today by NASA and the aerospace industry are based on decades of study of early spaceflight concepts. In our view these architectures are not an acceptable solution for a new lunar transportation system that will be required to support emerging lunar activities at reasonable cost. Genuine innovation is needed to achieve the goals of affordability and sustainability called for by the President.

LTS is developing a new lunar architecture concept [21] that, they believe, is better suited for a state-of-the-art lunar transportation system. This architecture is characterized by modularity and extreme flexibility leading to reduced development cost and better evolvability. A hard look at this architecture will show that it enables NASA to meet its strategic objectives, including sending small payloads to the lunar surface in a few short years, sending larger payloads to the lunar surface in succeeding years, and sending crews to the moon and back to the Earth by the middle of the next decade. The basic LTS vehicle is reusable, and the second flight to the moon is accomplished with new tanks and payloads launched from Earth. Figure 9 depicts the hardware transported to LEO on EELVs with an opening shroud and supported with a continuing flow of propellant tanks [21] and payloads for the support of lunar surface operations. The space stage shown in Fig. 9 is used in orbit, if possible.

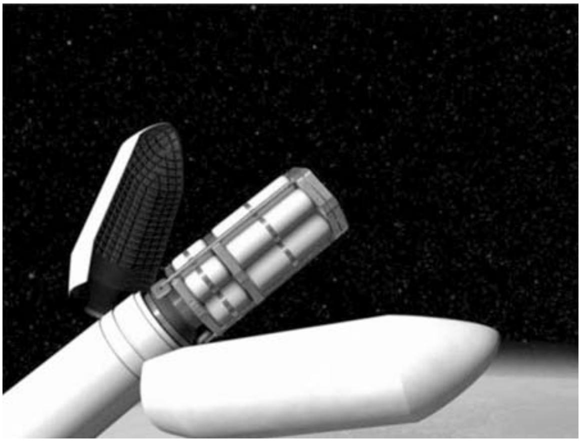


Fig. 9 Lunar Transportation Systems, Inc., proposes commercial expendable launch vehicles to transport its cargo logistics system to low Earth orbit and the LTS system provides services to and from the moon's surface.

C. Use of Existing EELVs or Ares Vehicles

This new lunar architecture utilizes current ELVs or EELVs to bring a new fleet of reusable spacecraft, lunar payloads, propellants, and eventually crews from the Earth to LEO. The reusable LTS spacecraft does the rest of the job. They take payloads from LEO to the lunar surface and bring payloads back to Earth from the moon. This architecture permits a “pay as you go” and a “technology development pathway” that allows NASA to achieve a series of its strategic objectives as funding and technology developments permit. The LTS approach reduces mission recurring cost by advancing in-space transportation technology, and later, lunar resource utilization, because this is much less costly than investing in new Earth to orbit (ETO) transportation. Figure 10 depicts the initial LTS mission to the lunar surface using full propellant tanks from LEO.

D. Lunar Payload Capabilities

The size of the payloads delivered to and from the moon depends on where and how many times lunar landers are refueled on their way to and from the lunar surface. The initial fleets of reusable spacecraft vehicles are sized to fit the payload capabilities of Delta II heavy class launch vehicles or a similar class of vehicle. This architecture is capable of delivering 800 kg to the lunar surface directly from LEO without the need to refuel in space. The LTS method is capable of delivering payloads of 3 metric tons to the lunar surface with refueling at L1 only. It is also capable of delivering 6 metric tons to the lunar surface with refueling at MEO, at L1, and in lunar orbit. Comparable payloads can be returned from the lunar surface to LEO or to the Earth by refueling the lunar lander at one or more of those

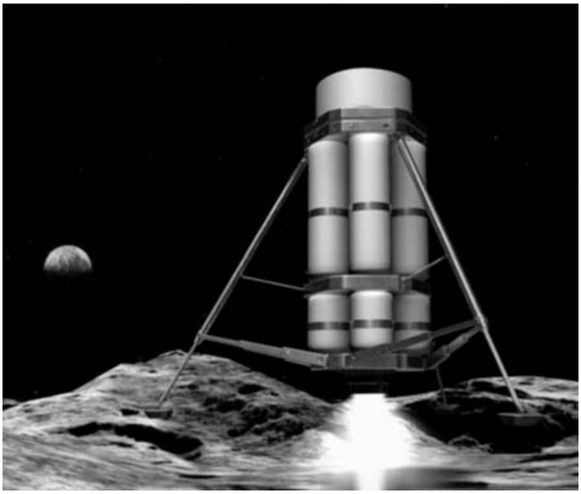


Fig. 10 Lunar Transportation Systems, Inc., lands 800 kg on the moon’s surface and adds propellant depots to mature the commercial cargo service to 10 tons using their reusable lunar landers.

locations. While this initial system is not meant to transport crews to and from the moon, LTS is meant as a technology development testbed for a crewed Earth-moon transportation system and initially transports only nonemergency cargo to and from the moon's surface.

E. Reusability

A key feature of this Earth-moon transportation system is that the two principal LTS spacecraft, the lunar lander and the propellant transporter, are reusable. The lunar lander transports payloads from LEO to the lunar surface and back. The propellant transporter transports cryogenic propellant tanks from LEO to any place in cislunar space where lunar landers need to be refueled. Figure 11 top, left to right, depicts the four basic elements of the LTS hardware: the propellant transporter, the vehicle with an engine, payload transporter, and the lunar lander. After the initial flight, the vehicle hardware is reusable and the tank transfer technology evolved to being capable of being refilled in space and on the moon.

F. Other Enabling Technologies

This state-of-the-art architecture does not depend on the development of any new heavy-lift launch vehicles. It does depend on the development of six emerging

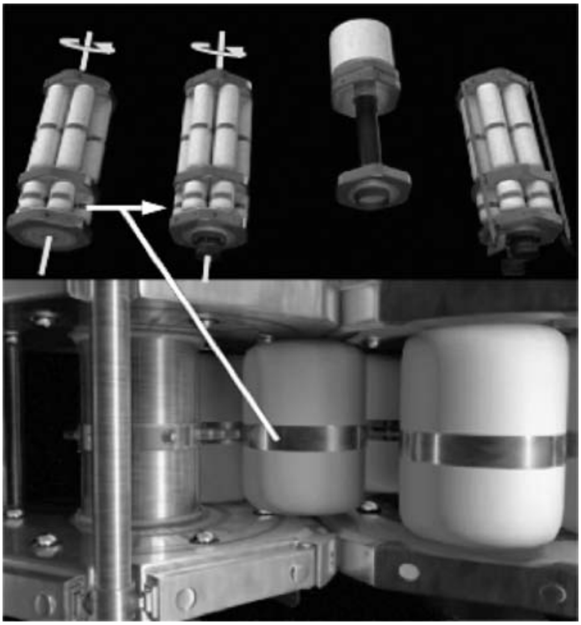


Fig. 11 Lunar Transportation Systems, Inc., transfers cryogenic tanks using telerobotics on the trip to the moon's surface and grows into reusable systems to provide cargo services.

technologies: 1) an autonomous rendezvous and docking system, 2) an autonomous payload transfer system as shown in Fig. 11, 3) a spacecraft-to-spacecraft cryogenic propellant tank transfer system, 4) an autonomous propellant tank tapping system, 5) an autonomous lunar landing system, and 6) an autonomous lunar payload offload system. Developing these technologies is less risky and less costly than investments in new ETO transportation or cryogenic propellant transfer technologies. These emerging technologies, except autonomous rendezvous and docking (AR&D), can be developed by ground test. The LTS program plan includes a flight demonstration program in LEO and early robotic missions to the moon to validate these technologies.

1. Scalability

This new lunar transportation system is scalable. A follow-on fleet of larger spacecraft, designed to fit the payload capabilities of Delta IV heavy class launch vehicles, can transport payloads of up to 30 metric tons from LEO to the lunar surface, depending on where and how frequently they are refueled on their way to and from the moon. These larger spacecraft will be capable of transporting crews to the lunar surface and returning them to the Earth. They will also have the capability to provide heavy cargo transportation to support a permanent lunar base. Figure 12 depicts the scalability of the proposed public private system from ~ 1 m diam for an ETO payload diameter of 3–5 m to a 10-m-diameter tank in the SDV and NASA exploration vehicles.

2. Methodology

LTS plans to develop a fleet of spacecraft that are sized to fit the payload envelope and the payload capabilities of Delta II heavy launch vehicles to validate LTS

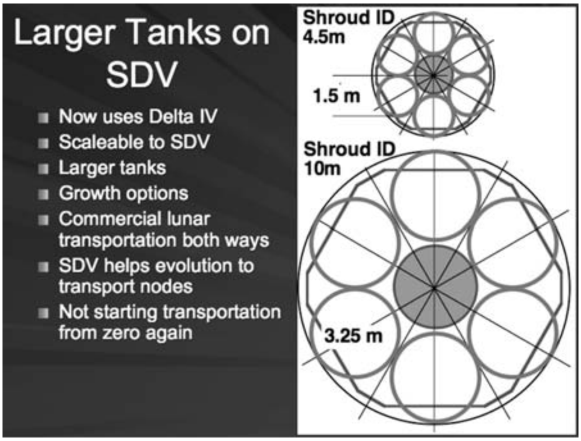


Fig. 12 Lunar Transportation Systems, Inc., offers scalability in payloads to orbit and the size of the tanks used in their reusable systems providing cargo services.

concepts and to deliver payloads to and from the lunar surface. Once LTS concepts are validated using Delta II heavy launch vehicles in a series of flight demonstration missions, LTS plans to develop larger spacecraft that are sized to fit the payload envelope and payload capabilities of Delta IV heavy class launch vehicles. The larger spacecraft fleet will have the capability to bring crews and heavy payloads to and from the moon.

3. *Crew Safety*

A very important element of the LTS lunar architecture is crew safety. Early mission are unmanned and prove the safety and reliability before crewed missions are attempted in commercial and/or government customers. Commonality of modules and subsystems increases flight operations experience rapidly, leading to greater safety. Backup lunar landers can be prepositioned at L1, in lunar orbit, or even on the lunar surface to provide crew rescue capability in case of a mission abort situation. Before any crew is transported, an appropriate number of successful unmanned flights will be accomplished to prove the system is safe and reliable enough for crew use.

4. *Cost Reduction*

The nonrecurring costs to develop this Earth-moon transportation system are much lower than the cost of developing systems that use more traditional architectures because there are fewer unique developments and it relies on existing launch vehicles. A significant reduction in lunar mission costs comes from the reusability of the major elements of the LTS system.

The largest cost for lunar transportation, perhaps as much as 70% of each lunar mission, is the delivery of the spacecraft, the propellants, and the lunar payloads from the Earth to LEO. The use of the larger payload diameters of the Shuttle Cm and/or the NASA CEV hardware to transport LTS hardware to LEO, could increased the effectiveness of the LTS concept by increasing the initial diameter of the LTS tanks. LTS will complete this phase using existing expendable launch vehicles. While these EELVs are expensive to launch, the development cost of significant new launch capability represents dozens of launches and many years of flight operations experience. When propellants can be produced on the moon, 60% of the operations costs may reduce Earth-moon mission return costs. If and when reusable Earth-to-LEO launch vehicles become available, a further 60% in costs or more may reduce lunar mission costs.

G. Schedule

Because this system relies on existing technologies and existing ELVs and requires only the maturation of several enabling technologies, it can deliver payloads to the lunar surface relatively quickly and well within NASA's schedule for robotic and human lunar exploration. As the LTS concept evolves, a propellant depot could contribute to the ability to store cryogenic tanks for a longer term in the hard vacuum of space and provide other benefits to LTS. The node system

could use the discarded LTS hardware and tanks to evolve into a future phase of increased reusability and expanded commercial operations using a system of nodes [20, 21].

H. Bottom Line

The LTS lunar architecture is based on concepts that reduce lunar mission life-cycle costs and technical risks, improve reliability and crew safety, accelerate lunar mission schedules, and allow for the routine delivery of lunar payloads on the equivalent of a two-way highway between the Earth and the moon. Commercial ventures like LTS, Inc., depend on the markets that can evolve from public private–partnerships that bring government markets to private solutions where the financing strength of the private sector is combined with the previous government market to create lower cost operations that create a tax flow to government rather than a drain on tax revenue.

V. Propellant Depot Public–Private Partnership Example

Whereas the LTS route to the moon does not initially require an established propellant depot or platform or transportation node in LEO, such a platform or node could use the discarded LTS propellant carriers and third stages and will probably logically emerge for other customers. The platform is an orbital location suited for a more permanent storage of the pre-filled LTS tanks.

A. Why a Propellant Depot?

LEO is much like a shoreline on Earth, where the over-land transportation vehicle requirements change dramatically to ocean transport vehicle requirements, and history tells us this usually creates a logical point for commerce to emerge like a harbor. When trade routes cross on land, sometimes a town emerges and supplies transportation services and grows as a result of the transportation. Similar evolution is likely in space with some adjustment for the differences of the rest of the universe. Figure 13 depicts a container ship loading at a port, which is the current result of 50,000 years of trade route evolution on Earth. We can learn from studying trade routes and their support functions. Commercial transportation cycles are defined as the most effective movement of mass from A to B using the best vehicle for the job. The containers transfer to and from a ship to land modes of transport including truck, rail, and other forms of transportation. An aerospace or aircraft company could argue that one could just fly the cargo and passengers and eliminate the port, but for 99% of the total cargo mass transported on Earth, that solution is just too expensive. An examination of other Earth remote bases that have been developed and used successfully seems to confirm that two or more forms of logistics transportation are required, depending on the mass to be moved. In Alaska, these forms of transportation cycles included passengers on 737 aircraft at \$5/lb, air cargo at ~\$1/lb, trucks at 10 cents a pound, unmanned barges at pennies a pound, and the most used pipeline, approximately one-tenth of a penny per pound. Thus logistics at remote bases is

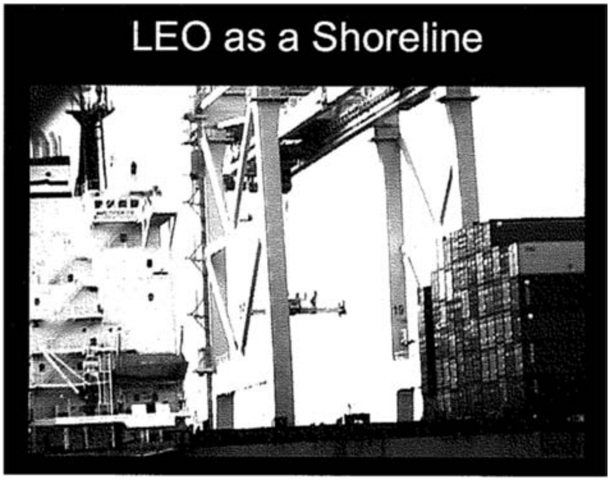


Fig. 13 Mature trade routes on Earth transfer containers at the shoreline and create a node, because vehicle requirements change, but space transportation still uses the same vehicle design through several vehicle requirement changes.

all about the economics of transportation, and the longer the logistics route or remote the logistics location, the more critical is the economics, and more drivers for more than one transportation solution is critical to the survival of the venture. To base the entire logistics transportation infrastructure on what humans cost to transport means you are not asking the right people for quotes. A 50-year sustainable logistics transportation system to and from the moon will include more than one transportation method with upgradable systems that cut the costs significantly as the market matures. For example, the pipeline transported more mass in the first 90 days than all the mass barged, trucked, and flown to Prudhoe Bay over a 10-year development cycle. When one looks at the trade routes on Earth, it is apparent that a certain commercial evolution occurs as the logistics route matures, and the most efficient vehicle consistent with cost and safety is used on each transportation leg of the route. Humans fly at increased cost and noncritical cargo moves on less reliable and less costly vehicles. The transportation cycles or trips we see in space transportation includes six separate lunar cycles in a lunar roundtrip [22]. The one-way trip changes vehicle requirements at LEO and LLO. These orbits are likely locations for nodes, because vehicle requirements change at those locations. In trade routes on Earth, for example, the transition from ocean to land has resulted in vehicle changes and an evolution to harbor locations. When the reader compares these six transportation cycles to current space shuttle operations and expendable vehicles, it becomes apparent that propellant depots and/or nodes will emerge or evolve between each cycle, and the nodes will provide transportation services to the vehicles like Earth trade routes. It should be pointed out that the moon's surface is 47 times more remote than the North Slope of Alaska with significantly more transportation risk.

B. Public–Private Partnership: Commercial Propellant Depot

The early propellant depot is stimulated by commerce and draws on private investment via the public–private partnerships. Selling propellant in LEO requires a constant supply of propellant deliveries, which create a multitude of commercial vehicles delivering everything from raw cryogenic propellant to pre-filled cryogenic tanks that require storage in the hard vacuum of space and without the warming rays of the sun or Earth radiation. Selling propellant requires the early customer and quickly evolves into servicing vehicles and transfer of payloads like any port or filling station now provides. The propellant depot quickly expands to other orbital inclinations with new transportation nodes, which sell propellant, but offer the expansion into other forms of commerce. The long massive discarded hardware platform makeup of these nodes lends themselves to propellant settling gravity, payload enhancement by momentum exchange, and tether operations. Tethers, for example, can facilitate the approach to an orbital facility, much the way a mooring line does for a ship approaching a dock. Reference

Just how could a commercial propellant depot evolve from a public–private partnership with NASA and other international government agencies on the public side and commercial organization from large aerospace companies to commercial space entrepreneurs on the private side of the partnership? The key is how to break down the barriers that exist without driving everybody away [3]. New Zealand broke down the barriers, and so we know it can be done once a crisis situation is realized, but how does the public–private partnership emerge without a crisis situation? Opportunities are starting to emerge, like the Centennial Challenge fuel depot at NASA, the lunar Roundtable, a group of organizations interested in lunar development in many diverse commercial forms and international opportunities to create markets using private money and the NASA COTS Procurement, which should create a flow of commercial companies willing to transport propellant at rates attractive to fuel depot operators.

One result of commercial organizations and the innovation they sometimes bring is the difference in the innovation and the approach to the marketplace. This diverse approach is really a series of approaches with market forces and private investors picking the best commercial concept and approach to put investment funds into and make it work. Not always does the best idea come to the top of this “market forces” environment, but with the government as a public member this risk is reduced. At any rate it is better for the government than paying the entire cost from the agencies budget, which is under increased pressure.

C. What Would a Propellant Depot Look Like in a Public–Private Partnership?

Propellant depots by NASA might approach the sophistication and cost of a portion of the space station. If built by commercial forces and private investment, then markets, initial cost before revenue flow, innovation, and automation are important.

What diverse approaches might be proposed? A commercial space venture of a non-critical cargo tries to place the customers first and profits second with maybe future growth markets third. If LTS, Inc., salvaged the discarded EELV hardware, then the orbital node after the first lunar transportation mission might look like

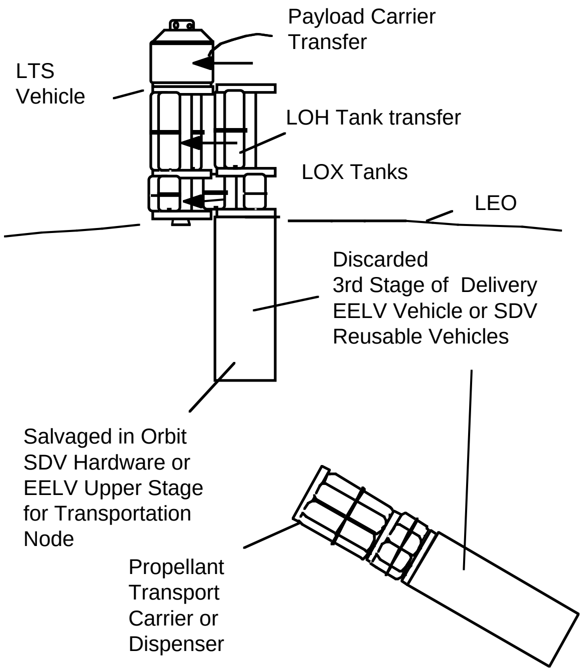


Fig. 14 Phase I propellant depot using spent stages of LTS payload delivery vehicles in low Earth orbit.

Fig. 14. It contains the discarded EELV third stage and the propellant carrier combined in a manner capable of servicing the next LTS mission. Each spent stage and other hardware represents “land” on an economic frontier, where real estate is expensive. This LTS customer requires a method of extending the cryogenic liquid storage duration in orbit, and so shading from the sun is an issue. A NASA customer might need a large propellant transfer for vehicles going beyond low Earth orbit. To accommodate this requirement, the affordable commercial vehicles would deliver propellant from the Earth’s surface to storage tanks at the propellant depot; the depot would provide increased cryogenic propellant thermal storage capability via innovation and fill the large tanks on the NASA vehicles. The problem of long-term cryo storage is mitigated by innovation and everything is excluded, except the technology capable of extending the long-term cryo storage for the least cost. Commercial market forces are at work, but the public partner would have the knowledge to choose the correct new technology that has the greatest benefit for the nation’s future and solve the near-term problem. The focus is more commercial by using technology to solve existing and future problems, not creating technology and looking for a place to use them.

Figure 15 depicts the next evolution of the commercial stage 2 propellant depot. It contains additional discarded EELV third stages and the propellant carriers plus other hardware combined in a manner capable of servicing the larger customer base.

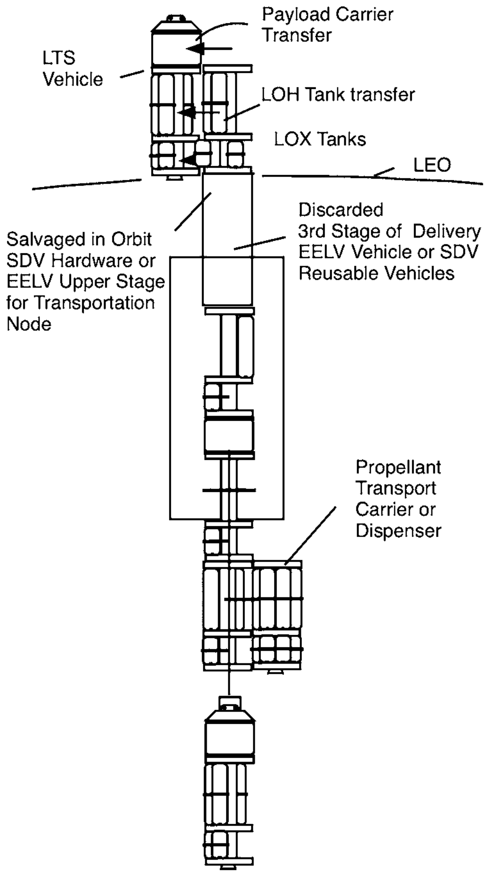


Fig. 15 Commercial propellant depot stage 2 using launch hardware and new components.

VI. Lunar Resource Development via Public–Private Partnership Example

Figure 10 depicts the LTS vehicle landing on the lunar surface. Once affordable routine transportation is available to and from the lunar surface, can prospectors be far behind? Figures 14 and 15 start the evolutionary process of building a sustainable lunar transportation system for non-critical cargo in both directions with lunar resource recovery for space solar power projects being some of that cargo.

A. Future Benefit Back to Earth

What could be valuable enough on the lunar surface to justify prospecting? Potentially, helium 3 can be mined on the moon and transported to Earth, probably

transported as a cryogenic liquid. Helium 3 is a product of our sun and transported to the lunar regolith via the solar wind over millions of years. Helium 3 is deposited on or near the lunar surface and tends to get mixed by impact objects striking the moon. It is thought by some experts to be a compound that could be used in a fusion reaction to produce power without the radioactivity by-product typical in today's nuclear power plants. America generates approximately 20% of our grid power using nuclear power plants. France generates over 50% of their grid power using nuclear power plants. At $60/barrel for oil, the energy equivalent of a ton of helium of 3 is about $8.7 billion per ton.

If the lunar surface requires sustained development, then reliable and affordable transportation in both directions will be essential. The transportation starts in a modest manner from LEO and expands on the growth of lunar markets and the value of lunar resources to mankind on Earth. It appears logical that outposts or transportation nodes around each new celestial body are inevitable in the exploration process beyond Earth, just like harbors have been inevitable to mankind's search and development of resources on Earth. Gold, diamonds, food, oil, and various ores have been the focus on Earth for resource recovery. The resources are sold, profits made, and industries created, much the way all commerce starts.

Nodes will emerge between destinations, and a mature lunar transportation system will evolve and have a sustainable revenue cargo in both directions to permit what we build in the next several decades to survive well over many centuries. The transcontinental railroad started with existing technology and upgraded on profits, lasted 150 years, and is just now being overshadowed by trucks with a subsidized roadbed and aircraft with subsidized nodes.

The use of mass from spent vehicles and used cargo containers can be used to greater advantage the further the remote location is from the origin. Most traditional aerospace people cringe at the idea, but this reuse continues to be an effective tool in remote resources locations on Earth including the North Slope of Alaska.

B. Lunar Transportation Node System

The object of the lunar transportation node system is to provide a method and an apparatus for more efficient and more affordable transport, storing, transferring, and enhancement of payloads and their vehicles in space using an evolving transportation node system.

A primary object of the lunar transportation node system is cost reduction over an extended period of time over the total development of another celestial body by building a lunar "highway" made up of separate transportation cycles capable of being applied to other celestial bodies in the universe. The transportation of cargo between locations in space can be separated into a series of transportation cycles as has happened historically in transportation on Earth. A transportation cycle is defined as mass that starts, changes location, and stops. This means start and stop between the transport function is a node, which connects the various cycles. The node involves changing transport vehicle hardware, refueling, cargo transfer, and commerce. In mature space transportation cycles this means each individual transportation cycle of mass can use the most effective and least cost method of transport hardware. When the reader thinks about it, this maturing also results in the most effective vehicle on each cycle on Earth. Imagine the cargo container

originating overseas, it starts on a truck to the seaport, is loaded on a container ship, transported to another port, loaded on a train, and eventually finds its destination on a truck with vehicle servicing occurring in the background. In space, the node is a service station, cargo transfer and storage and the location where other commerce will evolve. Space transportation cycles existing now include a movement of mass from Earth to orbit with the vehicle discarded and no return trip. Limited round trip capability is offered by the mostly reusable space shuttle and fully reusable vehicles are anticipated in the future. Starting from Earth and accomplishing a round trip to the moon, for example, can be done, but has been expensive. The “one-shot” Apollo missions were expensive partly because all of the hardware was expended on each mission, and the propellant was carried from Earth, and most of the propellant mass required for the entire round trip was carried for most of the trip. High costs create a barrier to the commercial transportation hardware development of space and the investment of private capital in technically viable space transportation ventures. Part of this cost is the expense caused by expending the hardware and part is the logistics operations required. Propellant is currently 9/10 or more of the logistics mass beyond Earth orbit. This node changes that fact in an attempt to lower costs.

A primary advantage of the PPP proposed is a maturing of the Apollo one-shot expendable vehicle to a reusable transportation system with a capability of affordably moving mass in both directions using nodes to enhance the transportation process and stimulate commerce beyond our planet.

C. COTS and What Could a Public–Private Partnership Stimulate?

NASA has dedicated \$500 million to stimulating a commercial orbital transportation system (COTS) [23], but has limited the non-federal acquisition regulations (non-FAR) procurement to only new ideas and hardware, by limiting the use of the technology of the space shuttle or anything previously built and flown successfully. This limits the commercial innovation to new concepts and is like starting from zero once again. Mankind should be permitted to grow using the experience and success of the past. History has always been that way. Trade routes of the past once established were used again and evolved in some cases to interstate routes and autobahns. Sometimes modes of transportation changed and rendered the trade route as no longer the best, but limiting mankind’s innovation is a step in the wrong direction.

The SRB solid rocket booster supplies 71% of the propulsive energy to transport the space shuttle to orbit. SRBs should be the basis for the lunar vehicles and are the workhorse of the NASA exploration planning, but COTS was limited to not using shuttle heritage hardware, and the SRB and the external tank were excluded from consideration by the entrepreneurs participating by a last-minute NASA letter.

D. Future with Lunar Resources and What Could a Public–Private Partnership Stimulate?

The effect of the Earth’s gravity well is significant and is about 70% of the cost of a trip to the moon. Even most commercial advocates agree NASA should push

beyond the moon to other locations in the near universe. Lunar resource development is a little like the quest for the North Pole in the 1920s. Many countries attempted to be the first to the pole. The first success was a little like Apollo, but when commercial organizations entered the Arctic it was for other reasons, which included natural resources of value. The North Slope of Alaska, for example, required a number of oil companies to buy oil leases and gamble $12 billion of 1970 private-money dollars to develop one field in Prudhoe Bay, Alaska, when the oil wellhead price was estimated to be $6.95/bbl. Now 18 other fields have been developed around Prudhoe Bay, and another private investment, an $8 billion pipeline, transports all of the oil toward market. The North Sea oil is another example of recovering resources of high value under difficult conditions, when the economics justify the expense of recovery.

What forced these private risk takers to gamble in Alaska? It was the trillion dollars of oil to be recovered from the Prudhoe Bay field and the ability to spread the risk to multiple major oil companies and lots of smaller independents, plus a supportive public environment able to see the value for both sides in normal business practices stretched to fit the risk, the remoteness and private capital required. This kind of cooperative structure does not exist in the space business environment, partly because commercial people still struggle to get to the procurement table, where commercial markets might be available. The public–private partnership can change all that and more by giving the “second user” a chance to participate in the initial design to make it more appealing to commercial investors.

By 2066 all the farmable land on Earth will be under cultivation. In our children’s lifetime, off-planet resources will be important to mankind. After a sustainable lunar transportation system is established, lunar resources can be brought from the moon’s surface less expensively than up from Earth. These lunar resources can be important to the construction of space solar power facilities, transportation nodes, lunar produced propellants, and other commercial uses. Economics will force the use of lunar resources, and we as the human species should be doing everything in our ability to forward the development of this new space economic frontier and resource recovery opportunity. Lunar resources are critical to our future just like Prudhoe Bay oil was critical. The transportation infrastructure for lunar resource use is as important as it was to develop $6.95/bbl wellhead price oil at Prudhoe Bay, Alaska, when we could buy oil from the Arabs at $0.50/bbl.

Figure 16 depicts the expanded evolution of the commercial stage 3 propellant depot in orbit with the ability to expand into orbital markets related to space station and third-party orbital tug and other emerging commerce, because a node in one orbital inclination has been introduced into the space environment. The depot contains additional discarded EELV third stages and the propellant carriers recombined in a manner capable of expanded servicing including the length, mass, and tether capability for future lunar missions. Figure 16 also shows the effect of the combination of business organizations that result from public–private partnerships. The long gravity gradient node provides tether enhancement to payloads in both directions.

VII. Conclusion

Government space budgets are not alone sufficient to create the hardware, organizational infrastructure, and resource recovery settlements required to move

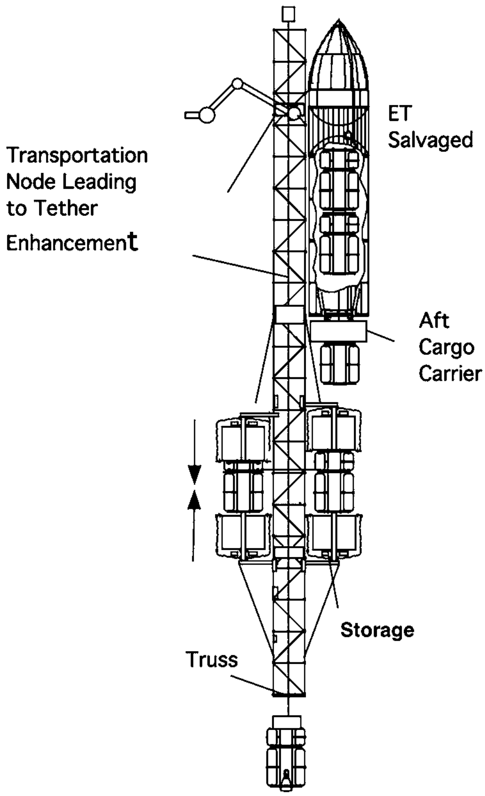


Fig. 16 Commercial propellant depot stage 3 using launch hardware and new components.

mankind off the planet quickly enough to find, develop, recover, and utilize the resources required from off-planet sources given Earth's population growth. The public–private partnerships [2, 3, 4] may help accelerate the cooperation between government and the private sector. New Zealand [1] has success in combining commercial operations with traditional government environments to create a tax rate decrease from 31 to 11%. COTS [23] could create the increased space budgets and the massive private investments required to accomplish space solar power [9], lunar solar power, helium 3, and other space projects [10] designed to benefit mankind.

References

[1] Chamberlain, A., "Tax Reform Lessons from New Zealand," http://www.taxfoundation.org/blog/show/600.html, New Zealand Herald, nzherald.co.nz.
[2] Osborne, S. P., "Public–Private Partnerships: Theory and Practice in International Perspective," Routledge Business/Economics/Finance, 2000, p. 348.

[3] "Public Private Partnerships," "Reminders," http://ncppp.org/, The Nat. Council for Public Private Partnerships, Washington, DC.
[4] "Richard Branson," http://www.virgin.com/aboutvirgin/allaboutvirgin/whosrichard-branson/default.asp, Virgin Galactic LLC.
[5] "Hubble's Future," http://hubblesite.org/newscenter/newsdesk/future/, Office of Public Outreach, Space Telescope Science Institute, Baltimore, MD.
[6] "Skylab History," http://www.xmission.com/~skylab/skylab.html, NASA-JSC, 1979.
[7] Taylor, Thomas C., "Heavier Lift Using Private Funds - Shuttle Cm," available from tctaylor@zianet.com.
[8] Griffin, Michael D., "NASA and the Business of Space," http://www.spaceref.com/news/viewsr.html?pid=18740.
[9] "Bright Future for Solar Power Satellites," http://www.space.com/businesstechnology/technology/solar_power_sats_011017-1.html [retrieved Oct. 2001], Leonard David, Senior Space Writer, 17 Oct. 2001, Space Solar Power "Two New Studies."
[10] Citron Bob, www.lunartransportationsystems.com, Lunar Transportation Systems, Inc., Bellevue, WA [retrieved Oct. 2005].
[11] Taylor, T., "Maximizing Commercial Space Transportation with a Public–Private Partnership," AIAA Paper 2006-5883, 2006.
[12] Darwall, Rupert, "Market Reform: Lessons from New Zealand," http://www.policy-review.org/apr03/darwall.html.
[13] Ambrose, S. E., "Nothing Like It in the World: The Men Who Built the Transcontinental Railroad 1863–1869," Touchstone Book, Simon & Schuster, 2000.
[14] "Space Island Group's Tourist Hotel Design," http://www.spaceislandgroup.com/press-release.html, Space Island Group, CA.
[15] Kistler, W., Citron, R., and Taylor, T., LTS Inc., www.lunartransportationsystems.com, Gimarc, A., GLOBAL OUTPOST, Inc., Meyers, G., Space Island Group, www.spaceislandgroup.com, "Space Bases for Transportation Support," STAIF Conf., Albuquerque, NM, E02 Session "Space Bases"15 Feb. 2005.
[16] Taylor, T., Pittman, B., Gimarc, A., GOI, Las Cruces, NM, Anchorage, AK, and San Jose, CA, Spencer, J., Space Tourism Society, LA, CA, Favata, P., Anastrophe Design, Sunnyvale, CA, Meyers, G., Space Island Group, West Covina, CA www.spaceislandgroup.com, "Habitation Testbed for Space Tourism," STAIF Conf., Albuquerque, NM, E06 Session, "Space Tourism," 17 Feb. 2005.
[17] Matsumoto, Shinji., Amino, Yoshihiko., Mitsuhashi, Tohru., Takagi, Kenji., Kanayama, Hideki., "Cost Study for Space Tour," http://www.spacefuture.com/archive/feasibility_of_space_tourism_cost_study_for_space_tour.shtml, Shimizu, Japan.
[18] Taylor, Thomas C., GLOBAL OUTPOST, Inc., U.S. Patent 6,206,328, "Centrifugal Gravity Habitation Torus Constructed of Salvaged Orbital Debris," March 27, 2001.
[19] Kistler, W., Citron, R., and Taylor, T., Lunar Transportation Systems, Inc., Bellevue, WA, "System and Method for Transportation and Storage of Cargo in Space," U.S. Patent 7,114,682, 7 June 2004.
[20] Kistler, W., Citron, R., and Taylor, T., Lunar Transportation Systems, Inc., Bellevue, WA, "Platform and System for Mass Storage and Transfer in Space," U.S. Patent 7,118,077, 11 March 2005.

[21] Kistler, W., Citron, R., and Taylor, T., Lunar Transportation Systems, Inc., Bellevue, WA, "Platform and System for Propellant Tank Storage and Transfer in Space," U.S. Patent 7,156,348, 11 March 2005.
[22] Kistler, W., Citron, R., and Taylor, T., Lunar Transportation Systems, Inc., Bellevue, WA, "Space Transportation Node Including Tether System," U.S. Patent Application No. 11/232,932, filed 23 Sept. 2005.
[23] NASA COTS Procurement, http://procurement.jsc.nasa.gov/cots/, 2006.

Chapter 3

Case Studies in Prediction and Prevention of Failures

Sabina Bucher*

Northrop Grumman Space Technology, Redondo Beach, California

I. Introduction

THE Chandra X-ray Observatory, one of NASA's Great Observatories, is a space-based telescope designed to observe X-ray sources. General background information on the Chandra spacecraft and observing program can be found on the Chandra X-ray Observatory's public information website at http://chandra.harvard.edu. Like any telescope, Chandra must be able to point at many locations in the sky and to set up its instruments to correctly observe a wide variety of science targets. Moving safely and efficiently from target to target and ensuring that the vehicle is correctly configured when it arrives is at the core of Chandra operations. The Chandra Flight Operations Team (FOT) Mission Planning group uses several suites of software to assemble weekly schedules. Each schedule includes all of the maneuver and reconfiguration commanding required to perform each observation and to safely stow the instruments when the radiation environment is too hostile to observe. It is the mission planners who ensure that every observation and engineering activity is scheduled in accordance with the vehicle constraints and the observation requirements requested by the user. Vehicle constraints include restrictions on pointing, stored momentum, and component temperatures. The FOT Engineering group, along with members of the Science Operations Team (SOT), verifies that each weekly schedule meets all of the observation requirements and vehicle constraints. Additionally, FOT Engineering monitors the performance of the vehicle to ensure that all of the components are operating correctly and will continue to operate correctly.

---

*Chandra X-Ray Observatory Mission Planning Manager, Chandra Flight Operations Team. Member AIAA.

Each of the spacecraft subsystems plays a role in meeting the science requirements of the mission. The Chandra propulsion subsystem was designed to provide three main functions: orbit insertion, attitude control during early orbit activation, and momentum unloading. Liquid apogee engines (LAEs) provided orbit insertions, the reaction control system (RCS) provided attitude control during early orbit, and the momentum unloading propulsion subsystem (MUPS) provides momentum unloading. Figure 1 provides a diagram of the Chandra spacecraft with the thruster locations marked. The LAEs and RCS thrusters were deactivated at the completion of early orbit activation and checkout activities, but the feedlines and thruster valves do continue to contain fuel. The MUPS provides momentum unloading to this day, by pulsing thrusters located on the corners of the spacecraft bus. Three of the four thrusters are fired simultaneously, at varying pulsewidths, such that the appropriate ratio of momentum is unloaded in each of the three spacecraft axes. The RCS and MUPS thrusters use hydrazine fuel, fed to the thrusters through stainless steel tubes, referred to as propulsion lines. The propulsion subsystem is equipped with thermostats and heaters designed to prevent freezing the fuel. The temperatures of the propulsion lines and thruster valves are monitored through thermistors located at various locations within the subsystem.

Chandra does have autonomous momentum unloading capability. However, since an autonomous unload could occur during an observation and degrade the pointing accuracy, the system momentum is closely managed and maintained below the autonomous unloading threshold. To maintain low system momentum, all unloads are planned and executed by stored command sequence. Detailed momentum propagation and a duration prediction for each scheduled unload is performed for every mission schedule.

In late 2002, a planned momentum unload took approximately 60% longer to complete than was predicted. The unload occurred at a warm attitude for the thrusters, but did not show any unusual thermal characteristics. Analysis of the momentum data showed that the thrusters had performed nominally at the start of the unload, but that the thrust from one of the thrusters decreased as the dump progressed. A few days later, short, isolated firings of the thrusters showed nominal performance from all four. However, several subsequent unloads showed

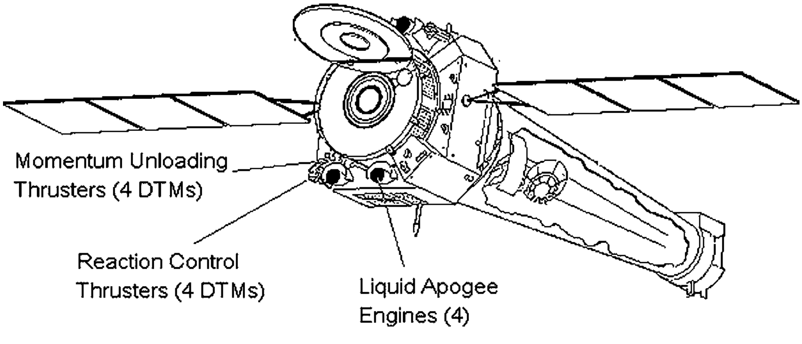


Fig. 1 Chandra spacecraft.

similar underperformance in two of the thrusters. Investigation of the anomaly led to the development of several analysis techniques and ultimately to a new constraint on the scheduling of momentum unloads. In the first case study, we will discuss one of the analysis techniques developed, which allows a pulse-by-pulse measure of the thrust provided by each thruster. We will also discuss the development of the scheduling constraint and the difficulties of incorporating such a constraint into the mission planning process.

Every maneuver changes the orientation of Chandra with respect to the sun, changing the distribution of light and shadow on the vehicle. As the amount of incident sunlight on each component changes, the temperature of the component changes. The propulsion subsystem is positioned around the forward section of the vehicle. Therefore, the further toward the tail of the vehicle the sun is (tail-sun attitude), the colder the propulsion subsystem components get. In early 2004, two propulsion line temperature sensors began to dip below their caution low limits. The violations were brief, infrequent, and only occurred at tail-sun attitudes. The change in behavior was initially attributed to changing scheduling constraints, which increased the frequency of tail-sun attitudes. Each limit violation showed that the appropriate heater was turning on later than desired when the spacecraft was maneuvered to a tail-sun attitude. Once the heater turned on, the lines recovered nicely and remained above the limit. As the violations became more frequent, a technique that isolated heater cycles and identified the temperature of the propulsion lines when the heater turned on was developed. The analysis uncovered a steady, mission-long trend in the propulsion line temperatures. Something had to be done to reverse it. The most straightforward solution would be to disallow the tail-sun attitudes causing cold temperatures. However, these attitudes were becoming increasingly important for keeping other spacecraft components cool, and eliminating them altogether would make some types of time constrained science observations impossible. Developing a new scheduling constraint would prove to be a balancing act between keeping the propulsion lines safely above the freezing point of hydrazine, while allowing for some time at tail-sun attitudes. In the second case study, we will discuss the simple techniques used to isolate the heater cycles, the methods developed to generate the new scheduling constraint, and the complexities of working it into the mission planning process.

II. First Case Study: Reduction in Thrust

A. Momentum Management

The Chandra mission scheduling software built into the ground system was designed as a “black box” that could ingest observation requests (ORs) and engineering requests (ERs) and output efficient mission schedules. An optimization routine with knowledge of spacecraft constraints would do most of the work of mission scheduling. In practice, the optimization routine did not always produce desirable results. Furthermore, within the first few orbits of science operations, one of the science instruments started incurring damage from low energy protons. This led to a new scheduling constraint that was not anticipated before launch, and, therefore, not incorporated into the optimization routine. At least in the short term, building safe and efficient schedules required manual intervention.

Unfortunately, a lack of visibility into the software made manually assembling a usable mission schedule an exercise in trial and error and thus an extremely inefficient process. To remedy the problem, the FOT designed a stand-alone tool that reads in observation and attitude requests and displays them in several frames of reference. The displays allow the scheduler to visualize the attitudes required for the week and to assemble a sequence of maneuvers and dwells (an attitude profile). The tool then displays the attitude profile with respect to various scheduling constraints, providing visual feedback of the impact of schedule changes on each constraint.

This tool, by integrating the attitude dependent torques contributed by the solar wind and the Earth's gravitational pull, predicts and displays the angular momentum stored in the spacecraft's reaction wheels for an attitude profile. With every schedule change the predicted momentum plots are updated and the scheduler gets immediate feedback on the impact of the change with respect to the momentum profile. The momentum plots facilitate using attitude to balance the solar and gravity gradient torques whenever possible. This reduces the required number of momentum unloads, minimizing both fuel use and thruster degradation.

When the system momentum does exceed the operational threshold, an unload is planned. The scheduler uses the momentum plot to place the unload in the most operationally desirable location. The tool then uses knowledge of the performance of the thrusters and the unloading algorithm used by the flight software to provide a prediction of the duration of the unload. Once the time of the unload has been selected, a call to a command sequence is added to the schedule at that time and the unload is added to the daily loads. Capability for automated scheduling of operational unloads was built into the mission scheduling ground software, but the simplicity and increased flexibility of scheduling unloads by hand has caused this capability to go virtually unused.

The Chandra MUPS thrusters are designed to be pulsed. The unloading algorithm used by the onboard flight program (OFP) fires the thrusters at a set firing period, but varies the pulsewidths. The pulsewidths are determined by rotating the required delta momentum vector into the thruster frame and then computing a ratio of firings required to unload the correct amount of momentum in each axis. A feedback loop is used to constantly update the delta momentum vector and thus the ratio of the pulsewidths. This algorithm produces roughly linear unloads from one momentum state to the next. The momentum unloading model in the FOT tool uses a simplified version of the unloading algorithm used by the flight software, but the model implemented in the mission scheduling ground software was designed to mimic the flight software as closely as possible. The OFP algorithm is replicated in the ground software, with a calibration of the in-flight performance of the thrusters modeling the feedback loop. The ground based algorithm has been shown accurate to within one firing cycle for a nominal unload. The thruster capability calibration, originally designed for predictive purposes, has proved to be invaluable in post-unload analysis.

B. Calibration of Thruster Performance

When generating the thruster performance calibration, the goal is to produce a thruster capability matrix, which provides the thrust per unit time produced by

each thruster in each axis. The flight data provides observed on-times and observed momentum changes. A least-squares fit of the observed change in momentum to the on-time data from a series of unloads can be used to solve for the thruster capability matrix. In this case, the data from 16 momentum unloads performed early in the mission were used to calibrate the early life performance of the MUPS thrusters. The unloads were selected such that each thruster was well represented in the cumulative data set and the unloads were performed at a constant attitude (not during maneuvers). The resulting thruster capability matrix produced an accurate estimate of the expected change in momentum for any set of thruster on-times. This calibration is used to provide accurate predictions of the thruster firings required and the total duration for any given unload and to compare the observed firings to early life performance.

C. Observed Reduction in Thrust

In late 2002, a momentum unload took approximately 60% longer to complete than was predicted. The unload occurred at a warm attitude for the thrusters, but did not show the thermal characteristics of any known thruster failure modes. The momentum profile of the anomalous unload began with the expected linear change from the starting momentum to the commanded ending momentum; however, approximately 350 s into the unload, there was a break in the linear trend. Furthermore, the pitch momentum rolled over in the second half of the unload. Figure 2 provides a comparison of the momentum states for a nominal unload and this first observed anomalous unload.

A few days later, short, isolated firings were used to assess the safety of using the thrusters and to check their performance. The thruster calibration used in the ground software was used to calculate the nominal dynamic response from these isolated firings. The result of each firing was compared to the expected values. The test firings indicated nominal performance from all four thrusters, which allowed for scheduling of further unloads. The subsequent unloads were closely monitored and the data from them compared to the expected change in momentum. Several of the subsequent unloads showed the same type of underperformance observed in the first anomalous unload.

The detailed momentum unloading model built into the ground system and the thruster calibration were used to simulate the conditions of the first anomalous unload. A series of simulations showed that the model best matched the observed behavior when the thruster capability for one of the sun-side thrusters was scaled back 300 s into the unload. This was very valuable information and did aid in the development of a fault tree; however, it was clear that a better data analysis method was required. The technique developed provided a pulse-by-pulse look at the performance of each thruster as compared to the thruster capability matrix. The technique was termed “thruster efficiency” because it shows the thrust produced as a percentage of the nominal value.

D. Thruster Efficiency Calculation

A tool that could provide the fidelity required for a pulse-by-pulse look at the progress of a momentum dump was crucial to the investigation of the anomalous

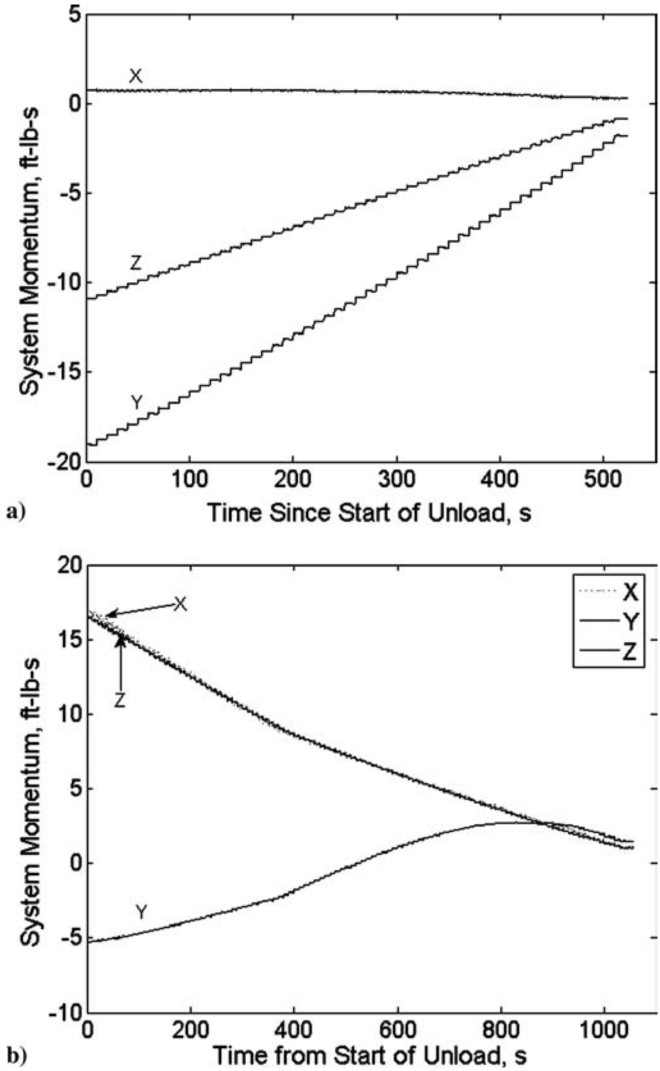


Fig. 2 Example momentum unloads. a) Nominal momentum unload. b) Anomalous momentum unload.

unloads. Such a tool would require very accurate measurements of the on-times and delta momentum for each firing period. With accurate on-times for each pulse, the thruster capability matrix, detailed in Sec. B, can be used to compute the expected change in momentum. A high-fidelity measurement of the observed change in momentum would then allow a comparison of nominal performance to observed performance.

The thruster on-times are available in the spacecraft telemetry stream, and so obtaining accurate, pulse-by-pulse on-times did not pose a problem. The momentum telemetry, however, did not provide the required fidelity. The first problem was the rate of the telemetry. The telemetered system momentum is downlinked at a rate only slightly faster than the thruster firing period. During an unload the momentum is changing rapidly, making the timing of the momentum measurements and the thruster firings very important. To increase the rate of the measurements, the momentum was computed on the ground using gyroscope and reaction wheel data, both reported more frequently than the telemetered momentum data. This increased the rate of the momentum measurements to approximately 40 times per firing period. With the increased rate of the momentum measurements, the dynamic response of the vehicle to the thruster firings became evident. Each thruster pulse imparts an impact to the vehicle. The impact causes a change in the spacecraft momentum, which is then compensated for by the reaction wheels, changing the momentum stored in the wheels. This is how momentum unloading works. However, the impacts also cause flexible components on the vehicle, most notably the solar arrays, to flex. With each pulse, flexing modes are excited, and momentum is passed back and forth from the wheels to the spacecraft body. This causes an oscillation in the reaction wheel momentum. The amplitude of the oscillation is often greater than the change in momentum produced by one firing cycle. Therefore, if the oscillation is not somehow compensated for, it will make the delta momentum measurements virtually unusable. Initially, a fast Fourier transform was used to solve for and remove the oscillation caused by the flexing modes. This process proved to be very manual and time consuming. The desire for a tool that could quickly and easily process many unloads called for an easier solution. The period of the flexing modes depends on the angle of the solar array, but is approximately one-third of the thruster firing period. This allowed a simple averaging of the momentum across each firing period to adequately account for the flexing modes. The averaging technique could be automated and produced a momentum solution with sufficient fidelity to generate accurate delta momentum measurements for each thruster pulse. Figure 3 shows the telemetered momentum measurements, the computed momentum measurements, and the computed momentum measurements once the averaging scheme is applied.

The final step to an accurate measurement of the momentum change imparted by the thrusters was removing the change in momentum caused by environmental torques. The algorithms used to propagate momentum for planning purposes were used to calculate the environmental torques over the course of the momentum dump. The calculated environmental torques were then subtracted from the system momentum to give a more accurate solution of the momentum change from pulse to pulse. Generating the compensated momentum profile, seen in Fig. 3, was the key pulse-by-pulse thruster performance analysis.

Figure 4 shows the computed thruster efficiency for the first observed anomalous unload. A nominal unload would show 100% for the entire duration. For this unload, the blue stars show that thruster 1, one of the sun-side thrusters, rapidly dropped to approximately 60% nominal thrust starting 360 s into the unload, and continued to decline for the remainder of the unload. As thrusters 3 and 4 completed unloading the momentum in their respective axes, fewer and fewer counts (shorter pulsewidths) were required. The signal-to-noise ratio of the calculation

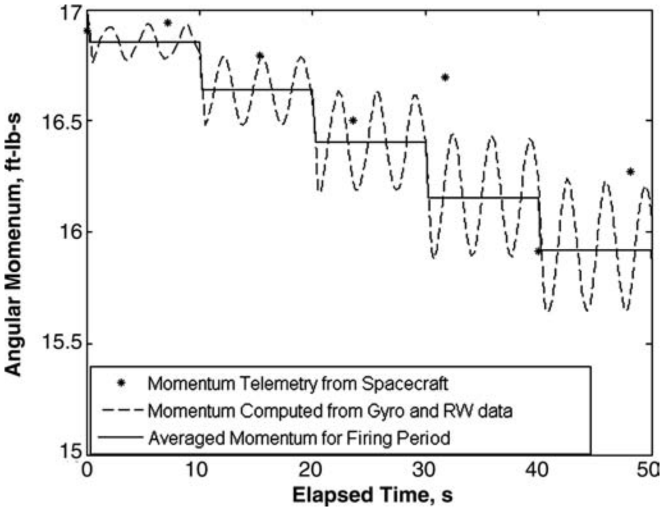


Fig. 3 Compensated momentum.

drops with the pulsewidth, which is why thrusters 3 and 4 appear to become noisy at the end of the unload. The clear picture of thruster performance the calculation provided was crucial to the anomaly investigation. The signature of anomalous unloads, demonstrated in Fig. 4, was used to rule out almost all of the possible failure modes identified in the early fault tree investigation.

E. Identifying Contributing Factors

Once the technique was completed, the thruster efficiency calculation was applied to all operational unloads. The resulting thruster efficiency plots were then checked for the anomaly signature seen in Fig. 4. Six unloads had exhibited the anomaly signature and three additional unloads showed indications of the anomaly but did not fully develop the reduced thrust level seen in the anomalous unloads (these were termed suspect unloads). When taken in chronological order, nominal unloads were interspersed with the anomalous unloads. The reduced thrust was not isolated to a single thruster; both sun-side thrusters exhibited the anomaly signature. This raised the question: What do the anomalous unloads have in common? It had been observed that the anomalous unloads tended to have high starting temperatures on the thruster valves. However, there were nominal unloads with equally high starting temperatures. All of the anomalous unloads had been longer than nominal unloads, but increased duration was caused by the anomaly itself. Looking at planned durations showed that the planned durations of the anomalous unloads was on the longer side for operational unloads, but there were nominal unloads that had longer planned durations than the anomalous unloads. So, was the anomaly caused by some combination of these factors, or something

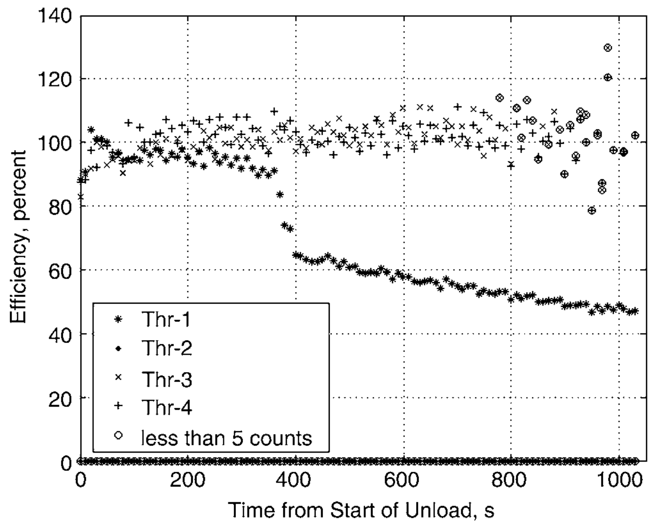


Fig. 4 Thruster efficiency of anomalous unload.

else entirely? Figure 5 shows the starting temperature of all operational unloads through 2003 on the *y* axis and the duration of the unload on the *x* axis. The markers indicate nominal, suspect, and anomalous unloads. Figure 5 implies that the combination of a warm starting temperature and a long unload duration led to the anomaly signature. Testing and analysis performed at the factory, which is not discussed in this chapter, showed that high starting valve temperatures combined

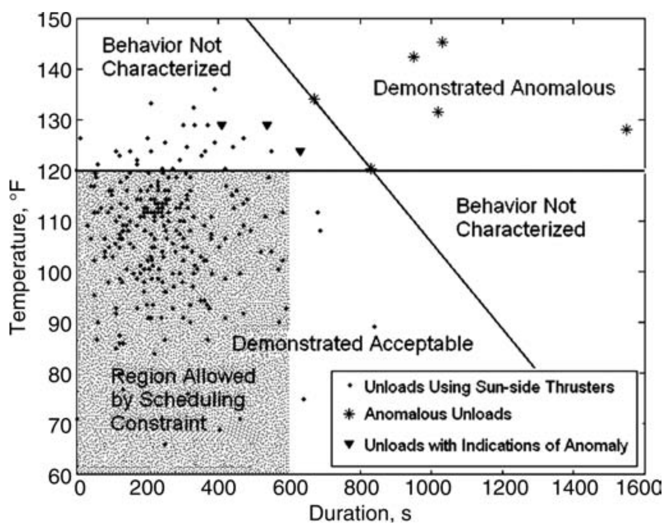


Fig. 5 Regions of demonstrated behavior.

with heat soak-back could produce the observed reduced thrust. Factory analysis also determined that infrequent firings in the degraded mode did not pose a safety risk to the vehicle, but that repeated, extensive operation in such a mode could lead to a catastrophic failure. This paved the way for a new momentum unloading constraint that prevented scheduling unloads with parameters demonstrated to contribute to the anomalous firings.

F. Defining and Implementing a Scheduling Constraint

With the data shown in Fig. 5, defining a new scheduling constraint was simple. The constraint became: Do not schedule unloads that are planned to be longer than 600 s or have starting valve temperatures above 120°F. As a safeguard, an automated shutoff was added for every scheduled unload. As discussed earlier, accurate estimates of unload durations were already available and incorporated into the planning tools. Implementing the automated shutoff required a simple modification to the command sequence that schedules an unload. Unfortunately, implementing the temperature restriction required knowing the starting temperature of the valves weeks in advance. This was not as simple.

At the onset of the anomaly, the relationship of the sun angle and the equilibrium temperatures of the thruster valves was well understood. If the sun was more than 140 deg from the spacecraft boresight, then the valves would eventually reach a temperature less than 120ºF. Of course, the time it took to reach 120ºF depended on the exact sun angle and the starting temperature of the valves. Initially, the mission planners worked with thermal and propulsion engineers to be sure that the vehicle was at a cold attitude for the valves for a sufficiently long time to allow settling to near equilibrium from any starting temperature. With a variety of other scheduling constraints driving the vehicle attitude profiles, it became highly desirable to reduce the dwell times required before an unload could be scheduled. To provide more accurate predictions of the dwell times required before unloading, a thermal model of the thruster valves was developed. The model is based purely on empirical thruster valve temperature data and was incorporated into the suite of mission planning tools developed by the FOT. The tools now provide a plot of the predicted valve temperatures over the course of the schedule, making it simple to identify acceptable times for unloads. Because of the lack of plotting capability and the algorithm updates required, the mission scheduling software in the ground system has not been updated to include the thermal model or to schedule unloads that meet the new constraint. There are currently no plans to make such an update.

With an accurate unload duration prediction, valve temperature model, and modified command sequence, it became not only possible but relatively straightforward to schedule unloads within the new constraint. The constraint has now been in place for over two years and has, to date, prevented the re-occurrence of the anomaly. By eliminating anomalous firings from nominal operations, the budget of isolated firings in the degraded mode can be reserved for contingency operations, when, for overall spacecraft safety, an unload must be performed regardless of thruster temperature. To date, no such contingency firings have been required.

G. Monitoring Thruster Performance

Before the reduced thrust anomaly, performance of the MUPS thrusters on Chandra was monitored by checking for unusually high or low thruster temperatures and making rough comparisons of the observed unloads and the predicted behavior. For many missions this rough, often qualitative, comparison of performance against expectation is where thruster monitoring ends. On the Chandra program, a tool that estimated the specific impulse (ISP) for each thruster was developed a few years into the mission. This tool was a step forward and, once completed, was run after every momentum unload. The ISP estimates are for the entire unload, and so they provide no insight into pulse-by-pulse performance. Therefore, unloads that behave nominally at the start and degrade as they continue do not always show a large enough change in total ISP to be detected. Since the thruster efficiency calculation was completed, it has been run on every operational unload. It is a valuable tool in monitoring for the re-occurrence of the anomaly, but is also an excellent tool for general thruster performance monitoring. For example, paging through thruster efficiency plots reveals a slight decline in the thrust per unit on-time over the mission. This trend is expected, but the ability to quantify it and monitor it for sudden changes (such as those seen with catalyst breakup that can lead to a wash-out condition) adds substantially to the probability of finding another subsystem anomaly before it becomes a failure. Similar tools would go a long way toward allowing other missions to detect thruster problems before they become thruster failures.

III. Second Case Study: Cold Temperatures on Sun-Side Feedlines

A. Thermal Protection of the Fuel Lines

The propulsion lines on Chandra are stainless steel tubing that provide fuel from the tanks to the thrusters. The lines are wrapped with heaters and multilayer insulation (MLI), designed to keep them safely above the freezing point of hydrazine (35ºF). Heater control is divided into circuits, each with a primary and redundant set of thermostats. Each set of thermostats controls power to all heaters on a given circuit; the thermostats are bimetallic disks, are not programmable, and cannot be commanded on. Propulsion line temperature telemetry is provided through thermistors located at various points on the lines. The thermistors provide telemetry only; they have no controlling function. During the spacecraft design phase, thermal models were used to place the thermostats and thermistors at what were expected to be the coldest portions of the lines.

Like many spacecraft, Chandra has a sun-facing side and an anti-sun side. The temperatures on the anti-sun side are relatively constant, but, because the vehicle is maneuvered several times a day, the temperatures on the sun-side can vary significantly. This is particularly true for the propulsion components, whose temperatures can change by over 100ºF depending on the orientation of the vehicle and the sun. If the sun is toward the boresight of the vehicle (low sun-pitch angle), the propulsion components are hot; if it is toward the tail of the vehicle (high sun-pitch angle), they get cold. When the sun-side propulsion components are in shadow, the heaters turn on, preventing the fuel from freezing.

Hydrazine cannot be allowed to freeze in the lines because it contracts as it freezes, allowing additional fuel to flow into the area vacated by the contracting fuel. If the hydrazine freezes solid, it can form a plug in the line, trapping the additional volume. Then, once the line heats up and the fuel begins to thaw, the expanding, thawing hydrazine has nowhere to go and causes the pressure in the line to increase until the plug breaks free or the line yields or ruptures [1]. A propulsion line rupture would allow venting of fuel and could contaminate the vehicle with hydrazine, a corrosive substance.

B. Cold Temperatures on the Sun-Side Fuel Lines

In early 2004, two propulsion line temperature sensors began to dip below their caution low limits. The daily minimum temperatures also showed subtle decreasing trends. The limit violations were brief, infrequent, and only occurred at high sun-pitch angles. A scheduling constraint implemented to prevent one of the spacecraft units from overheating had significantly increased the fraction of time spent at high sun-pitch angles. Initially, it was believed that the increase in time spent at tail-sun attitudes induced the trend in the daily minimum temperatures. When the limit violations started, it was felt that they could be explained by the physical separation between the thermostat and the thermistors. Figure 6 diagrams the layout of the thermistors and establishes the naming convention for the remainder of this chapter. The two thermistors showing low temperatures (thermistors B and C) are on the same heater circuit; both are located on the same corner of the vehicle, both on the sun side. There is a third thermistor (thermistor A) on the same heater circuit, located on the opposite sun-side corner of the vehicle. The third thermistor is very close to the thermostats controlling the heater circuit. This thermistor showed no limit violations.

As the frequency of limit violations increased, an analysis method that identified heater cycles and recorded the thermistor temperatures at each heater turn-on was run for all three thermistors. The thermistor near the thermostats showed a flat trend with time, but the other two showed decreasing, mission-long trends in the temperature at heater turn-on. Figure 7 shows the trend for all three lines. The changing thermal performance of the vehicle caused the portion of the lines monitored by the thermostats to warm with respect to the more remote section of lines. This

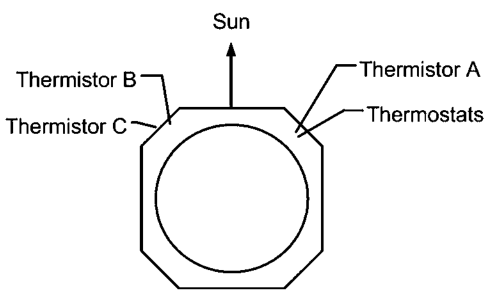


Fig. 6 Thermistor locations.

means that the thermostats controlling the heaters are no longer at the coldest section of the lines, possibly exposing portions of the lines to freezing temperatures.

C. Isolating Heater Cycles

Finding heater cycles and collecting the temperature at which the heater turns on can be performed by hand, but is a very tedious and time-consuming task. An automated routine that can isolate heater cycles and collect statistics is a far superior method. The increased computation and data handling capability of newer computers allowed implementing such a routine for all of the propulsion line thermistors on the Chandra vehicle.

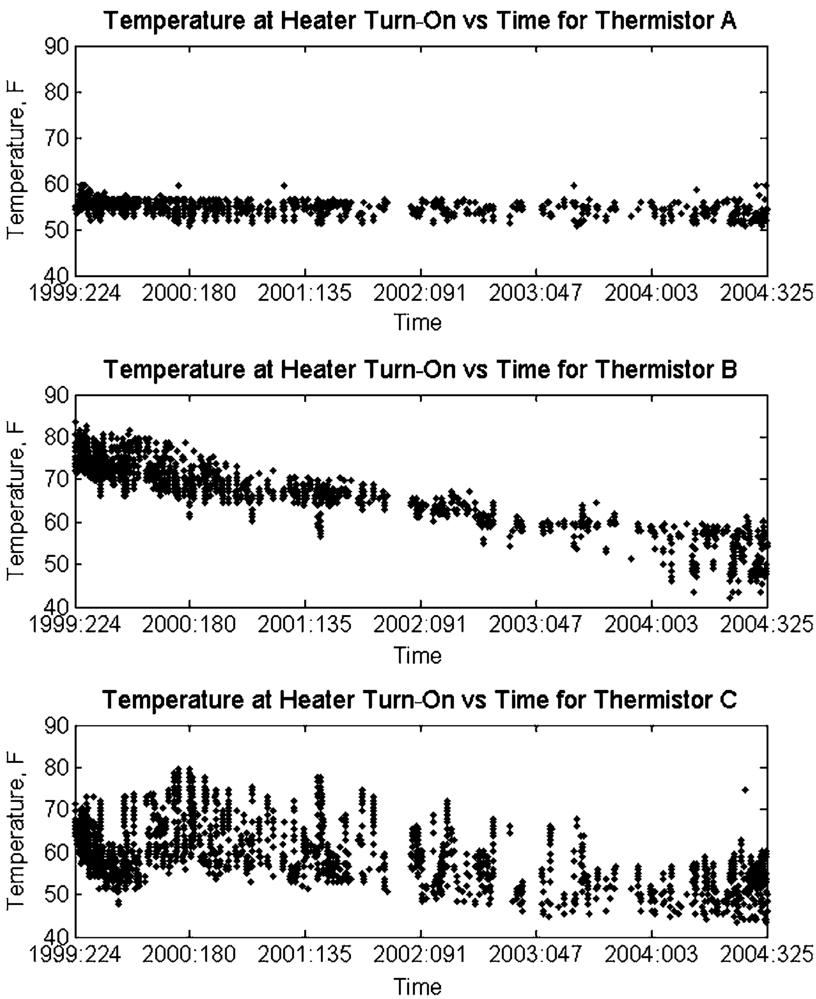


Fig. 7 Temperature at heater turn-on.

First, the temperature data were collected in pieces small enough to be easily manipulated, but large enough that several heater cycles were in a single block of data. Because maneuvers change the orientation of the vehicle to the sun and thus the temperatures of the propulsion lines, the temperature data were split by observation. This ensured that each piece of data had as consistent a sun angle as possible.

Once each block of data was collected, it was processed and the statistics for that attitude stored. The detailed telemetry data were then cleared from memory, before the next set was collected. Incremental data collection and processing drastically reduced the memory required to run the analysis over a long period of time.

The processing of each data set started with subtracting each temperature telemetry point from the one following it. This numerically estimates the first derivative, or rate of change, of the temperature. A heater turning on has a dramatic effect on the rate of temperature change. Once a heater turns on, a slowly cooling thermistor will suddenly show a rate of temperature increase that is too large to be caused by environmental heating. Therefore, setting an appropriate threshold for the rate of temperature increase allowed for automated identification of all heater turn-on instances.

Environmental heating and the quantization of thermistor data did, however, complicate determining and detecting the appropriate threshold. In this case, dividing the data into segments with a consistent sun angle helped to alleviate the complications of environmental heating. This was the first and most robust technique used for eliminating false positives due to solar heating. A boxcar average of the rate of temperature change over 10 measurements was used to account for the quantization of thermistor data. The boxcar average clearly identified a true rapid temperature change from a single toggle from one quanta the next. By requiring a sustained rapid rate of temperature increase, averaging also aided in preventing false positives. Plotting the boxcar average of the rate of temperature change for a period of time known to have heater cycles provided a clear threshold that was successfully used to identify heater on-times. Once the on-times were identified, the sign of the rate of temperature change was used to identify the exact turn-on and turn-off times for the heaters. With times as indices into the telemetry stream, all of the relevant statistics (e.g., sun angle, temperature) could be simply collected. It is this analysis that was used to find the trends seen in Fig. 7.

D. Temperature Behavior

The portions of the propulsion lines monitored by thermistor A, thermistor B, and thermistor C are the sections where the feedlines exit thermal blankets for access to the sun-side thrusters. Their locations put them in sunlight for forward-sun attitudes and in shadow for tail-sun attitudes. As the vehicle maneuvers away from the sun, the lines receive less direct heat input from the sun and therefore cool. This occurs for thermistor A, thermistor B, and thermistor C; however, they all cool at different rates and reach equilibrium at different temperatures. Because the thermostats are closest to thermistor A, it is the temperature near thermistor A that determines the coldest temperatures seen on thermistor B and thermistor C.

For most instances where thermistor A and thermistor B have reached equilibrium, thermistor B reaches equilibrium at a warmer temperature than thermistor A. However, during the initial portion of the cool down, thermistor B cools faster than thermistor A. This is due, largely, to the presence of the thermostats near thermistor A, which causes this section of line to decrease (and increase) slower than the section near thermistor B. At attitudes with sun-pitch angles greater than 170 deg this behavior becomes problematic. Thermistor B can cool sufficiently quickly that it approaches the freezing point of hydrazine before thermistor A reaches ~52°F (the turn-on set point for the thermostat controlling the circuit) and the heaters turn on. An example of this behavior is provided in Fig. 8. Figure 9 plots the temperature of thermistor B at heater turn on against sun-pitch angle, and verifies that all instances in the trend with time are caused by these high sun-pitch attitudes. The dip centered at approximately 173 deg sun-pitch shows that low temperatures on thermistor B are isolated to attitudes above 170 deg sun-pitch and that the cold temperatures are repeatable. It also shows that thermistor B does not reach a low temperature every time the sun-pitch angle is greater than 170 deg. Analysis of the temperature profiles for all attitudes above 170 deg sun-pitch showed that thermistor B reaches low temperatures when it starts out very warm, which occurs when the preceding attitude is forward sun. Thermistor B can fall uncomfortably close to the freezing point of hydrazine on maneuvers from warm attitudes to cold attitudes, due primarily to the difference in the cooling rates of thermistors A and B.

Unlike thermistor B, thermistor C is on a thicker propulsion line than thermistor A. This gives the line monitored by thermistor C more thermal mass, which causes it to react more slowly to rapid changes in environmental heating. Therefore, the maneuvers that cause problems for thermistor B are generally safe for thermistor C. However, thermistor C is generally colder than thermistor A, and there are

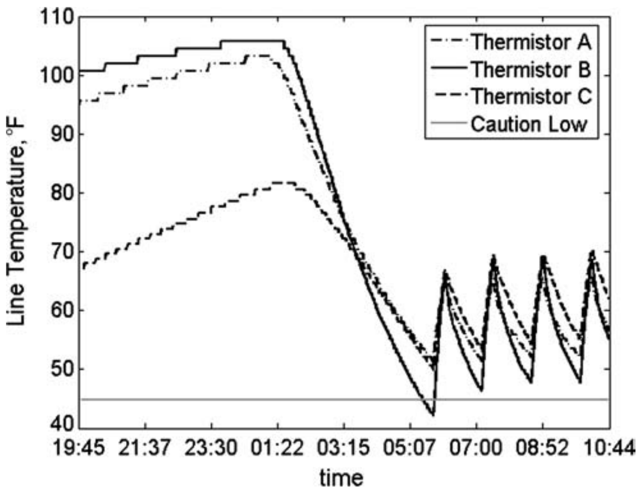


Fig. 8 Thermistor B problem case.

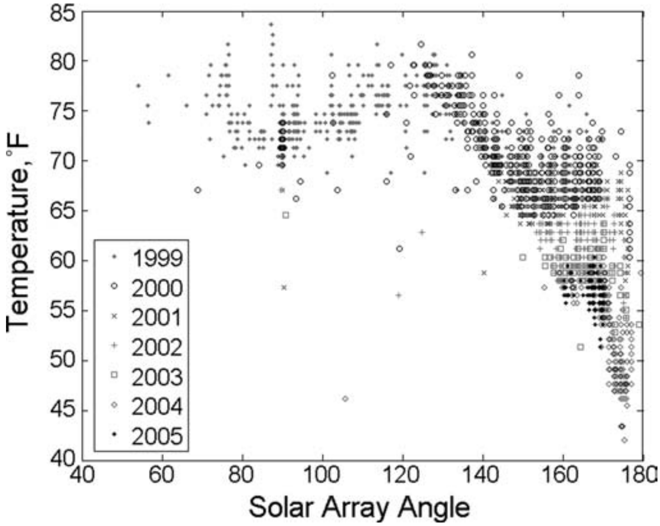


Fig. 9 Thermistor B temperature at time of heater turn-on vs solar array angle. (See also the color figure section starting on p. 645.)

attitudes where thermistor A reaches equilibrium above the heater set point and thermistor C reaches equilibrium below the acceptable operating range of temperatures for the propulsion components. An example of this behavior is provided in Fig. 10. Figure 11 plots the temperature of thermistor C at heater turn-on against sun-pitch angle as a means of identifying the problem attitudes for thermistor C. Figure 11 shows that thermistor C can reach cold temperatures at a wider range of attitudes than thermistor B. Analysis of the temperature profiles for all attitudes above 150 deg sun-pitch showed that thermistor C reaches low temperatures 1) on maneuvers from cool attitudes to cold attitudes (e.g., maneuvers from 140 to 170 deg sun-pitch) or 2) during long duration dwells at attitudes where thermistor A reaches equilibrium above the heater set point and thermistor C reaches equilibrium at an unacceptably cold temperature. Thermistor C can fall uncomfortably close to the freezing point of hydrazine at attitudes above 150 deg sun-pitch, due primarily to the difference in the equilibrium temperatures of thermistors A and C.

E. Defining and Implementing a Scheduling Constraint

Because the thermostats cannot be commanded to a higher set-point and the heaters cannot be commanded on, attitude is the only method that can be used to prevent cold temperatures on the sun-side Chandra propulsion lines. Therefore, attitude restrictions designed to keep both thermistors above 45ºF were developed and implemented. A constraint stating that no attitude hold would be scheduled with a sun-pitch angle above 150 deg went into effect on November 22, 2004. This 150 deg sun-pitch constraint made it impossible to cool the thruster valves

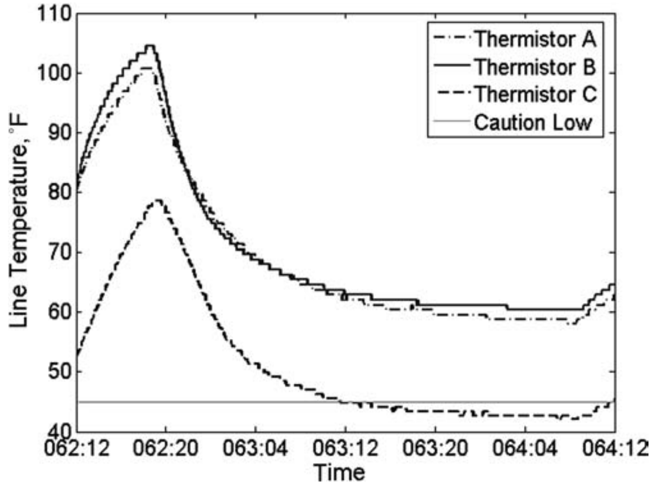


Fig. 10 Thermistor C problem case.

sufficiently to meet the guidelines put in place to prevent reoccurrence of the MUPS thruster anomaly, prevented certain types of science observations, and, due to reduced capabilities for cooling other spacecraft components, significantly reduced scheduling efficiency. For these reasons, there was a strong desire to relax the constraint without compromising vehicle safety.

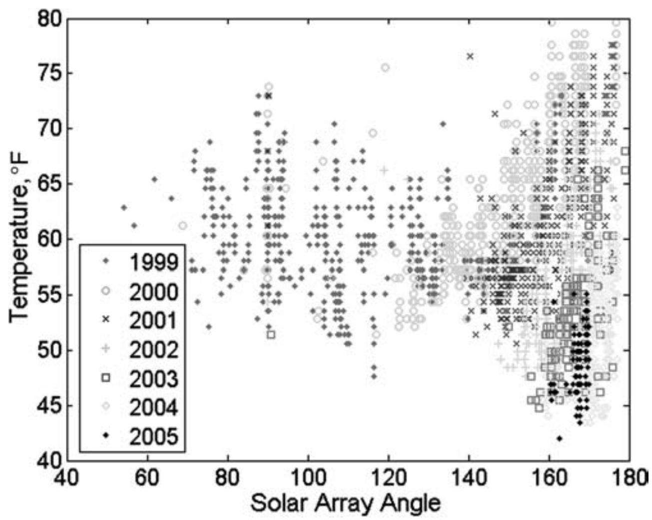


Fig. 11 Thermistor C temperature at time of heater turn-on vs solar array angle. (See also the color figure section starting on p. 645.)

Thermal analysts from the factory were asked to verify that the thermistors were measuring the coldest portions of the lines and to investigate the accuracy of the thermistors themselves. This investigation showed that the limit had some margin and could be lowered to 42.5ºF. With the new limit some attitudes past 150 deg sun-pitch could again be allowed. A study of the temperature profiles for all attitudes past 150 deg sun-pitch occurring in the previous two years was used to decide which attitudes would and would not be allowed. Since thermistor B and thermistor C behave differently, the constraint was split. The analysis of observed temperature profiles showed that 1) to ensure that thermistor B is kept above 42.5ºF, no maneuvers to attitudes past 170 deg sun-pitch can be allowed; and 2) to ensure that thermistor C is kept above 42.5ºF, no maneuvers to attitudes past 168 deg sun-pitch can be allowed. These new constraints went into effect on 8 December 2004.

The thermistor B constraint has been very effective in preventing cold temperatures on thermistor B. Recent data show that thermistor B has been above 50ºF at heater turn-on since the implementation of the constraint. The minimum temperature has also been above 50°F since implementation of the constraint.

The original attitude restrictions put into place for thermistor C had the same goal as the thermistor B restrictions: prevent cold temperatures before the heaters come on. However, as mentioned earlier, the problem area for thermistor C is not as distinct as for thermistor B. Additionally, as restrictions are put into place, mission scheduling changes in response. Primarily because of the desire to cool other spacecraft components, there is a bias in the way targets are selected favoring tail-sun attitudes. Therefore, as the restriction is pulled in to lower sun-pitch angles, more time is spent at the edge of the constraint. As this occurred, it was seen that thermistor C could drop near or below 42.5°F while the temperature of the thermostats stabilized above the turn-on set point for the heaters. As more time was spent at the edge of the constraint, more data were collected. Some of the new data required revising the constraint. As the constraint evolved and more data were collected, more instances of cool temperatures on thermistor C were found. As more instances were found, the constraint became increasingly restrictive, threatening to reintroduce the problems of the original 150 deg sun-pitch constraint.

In response, attempts were made to develop an empirical model like the one used to predict thruster valve temperatures. However, the temperature data set proved too sparse and too complex for an accurate model. The temperatures reached do not depend solely on sun-angle and starting temperature. The profile of attitudes leading up to the time of interest plays a significant role in determining the rate of temperature change. Two seemingly identical attitudes with respect to the sun can have substantially different temperature curves. Additionally, the time constants are relatively long, so that the lines rarely reach equilibrium temperatures. This introduces errors into generating a curve of equilibrium temperatures against attitude. Because an empirical thermal model is based on heat change rates and equilibrium temperatures, and neither can be determined to within an acceptable margin of error, the model will not perform as required. Therefore, another method of characterizing the behavior of the temperatures must be used.

The goal of the modeling effort was to allow some time in a region known to allow unacceptably cold temperatures on the propulsion lines without allowing

the lines to reach the low temperatures, in other words, to predict how long the vehicle can safely stay in the cold region. Establishing a worst-case cooling rate would allow safely spending time in a cold region without risking freezing the lines or requiring an accurate thermal model.

F. Determining Maximal Cooling Rates

To establish the maximum allowable time in cold regions, the line temperature data were analyzed with a goal of predicting the worst-case cooling rates at different attitudes. The sparsity of the data and the many factors that influence the cooling rate of lines makes an attitude-by-attitude analysis impossible. Therefore, the data were analyzed for attitude ranges, and worst-case cooling curves were established. The analysis used slightly over one year of data, broken into individual attitudes. The attitudes were sorted into bins by their sun angle. Within each bin, the attitudes were sorted by their starting temperature. The attitudes were then stepped through from hottest to coldest. The cooling curve started as the cooling rate from the first (hottest) attitude. The next attitude was then placed on the cooling curve at its starting temperature. If the second attitude cooled faster than the first, this became the cooling curve; if not, the cooling curve remained unchanged. Figure 12 shows a cooling curve after four attitudes have been analyzed. Figure 13 shows the completed curve with all of the attitudes used to compute it. This analysis provided a maximal cooling rate for all observed temperature ranges.

The final cooling curves were used to set the duration limits for various pitch regions. The curves were used to find the worst-case time to cool from a given starting temperature to 42.5°F. The constraint was set at 75% of this time, to account for missing data and possible changes in behavior. Plots of thermistor C vs sun-pitch angle over time were used to determine the pitch angle and durations

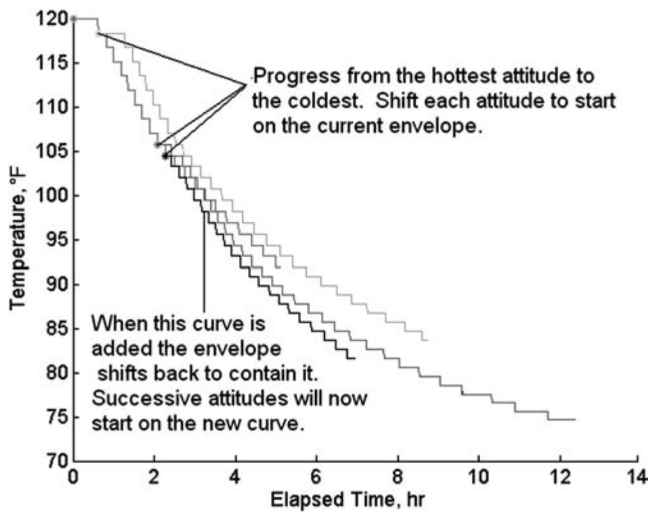


Fig. 12 Cooling curve generation.

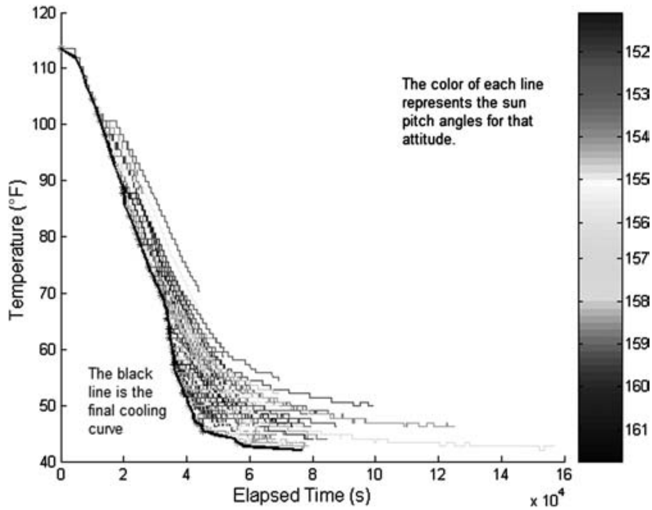


Fig. 13 Final cooling curve. (See also the color figure section starting on p. 645.)

required to deliver the starting temperatures listed in the constraint. These pitch angles and durations became pre-heating requirements before maneuvering to a cool attitude for thermistor C. The composite constraint, summarized in Fig. 14, was put into place on 23 June 2005, and has succeeded in preventing temperatures below 42.5°F on thermistor C since.

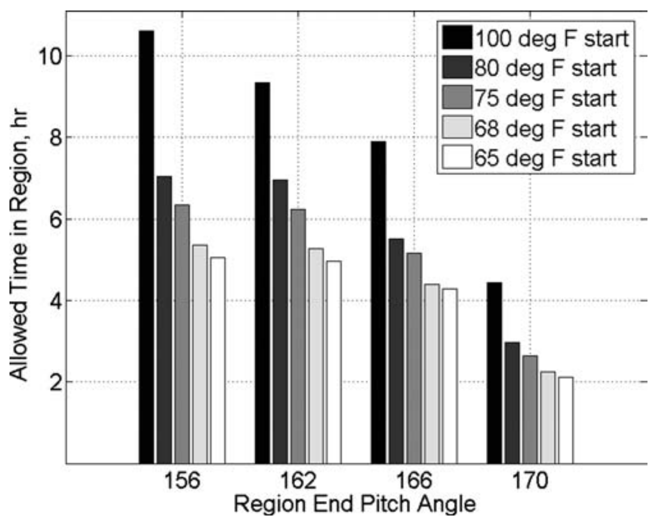


Fig. 14 Propulsion line scheduling constraint.

G. Implementing a Scheduling Constraint

With the constraints discussed in Secs. E and F defined, the problem became allowing efficient mission scheduling that met the constraints. The thermistor B constraint, do not schedule attitudes past 170 deg sun-pitch, is easily implemented. However, the thermistor C constraint is far more difficult. It requires pre-heating the lines before maneuvering to a restricted region, a simple enough concept, but the required pre-heating depends on both the attitudes used to pre-heat and which region the tail-sun attitudes fall into. To reduce the level of effort required to schedule within this constraint and to verify that it is met, a visual interpretation of the constraint was incorporated into the suite of mission planning tools developed by the FOT. The tool now provides a plot of the sun angle marking the time in the various regions. If the constraint is violated, the tool shows a red warning. With this visual representation the effort required to schedule within the new constraint became manageable. Primarily because of the algorithm updates required, the new constraint has not been incorporated into the ground system's mission scheduling software. There are currently no plans to make such an update.

IV. Conclusion

Both anomalies addressed in this chapter were detected and investigated through careful analysis, beyond traditional trending. The increasing availability of COTS analysis tools and powerful computers has made this type analysis feasible for routine operations. A least-squares fit to thruster performance over a series of unloads, allowing very accurate measures of nominal thruster performance, is now relatively simple to generate. As the Chandra MUPS thruster anomaly demonstrated, accurate measures of nominal performance allow earlier detection of off-nominal performance. Along the same vein, accurate measures of thermal performance, even when heater cycles or environmental heating dominate the trending data, can show changing thermal performance before it becomes a problem. Running though large data sets, finding heater cycles, no longer has to be a tedious, manual process. Incorporating analyses such as these, which look at the trends within the trends, can identify problems before they become safety risks, when there is still time to mitigate the contributing factors.

Mitigating the anomalies addressed in this chapter required rapid changes to the mission scheduling process. The scheduling process requires balancing many factors, such as maneuver size, momentum accumulation and unloading, mechanism motions, thermal management, and attitude restrictions. Such a complex problem leads people to want to develop an algorithm that does all of the work, a "black box" scheduling system that ingests requests and outputs a schedule with no manual intervention. However, such a system simply cannot support rapid changes to constraints and priorities. Adapting mission scheduling to evolving constraints, such as the ones required to mitigate these anomalies, requires a method that allows specifying how requests are scheduled. To prevent such specifications from being trial and error, the system must provide the user with the information required to make scheduling decisions. The Chandra mission planners use visual representations of the schedule and the constraints to assemble a set of maneuvers and dwells that achieves the scientific goals of the mission,

without compromising vehicle safety, in the most efficient manner possible. Optimization routines can be very helpful in solving the complex problem of mission scheduling, but they cannot be the only answer. A scheduling algorithm cannot work with engineers as constraints evolve, and cannot rapidly change its priorities and rules the way the human brain can. Adaptive mission scheduling requires a balance of automated tools and human input. Providing the scheduler with all of the information that goes into scheduling and using optimization routines to aid, not replace, the scheduler allows mission scheduling to evolve as the vehicle ages.

Like any program, Chandra has had its share of problems that were unanticipated before launch. Careful trending and thorough analyses have generally brought troublesome trends and changes in behavior to light before they have posed a threat to the immediate safety of the vehicle. To date, a majority of the problems experienced by Chandra have been successfully mitigated by modifying mission scheduling. The anomalies addressed in this chapter are examples of the type of analysis and creativity that goes into defining and implementing each new scheduling constraint. The tools and techniques developed during the operational portion of the Chandra mission have allowed the mission scheduling process to adapt to each new constraint. Without an environment that fosters in-depth analysis and an adaptive approach to mission scheduling, the Chandra mission could not have achieved the remarkable safety and efficiency record it has enjoyed to date.

Acknowledgments

The author thanks the following individuals for their contributions to the investigations, analyses, and tools detailed in this chapter: William Davis (Thruster Calibration and Thruster Efficiency), Tan Trinh (Thermal Models), Paul Goulart (FOT Scheduling Tools), Eric Martin (FOT Engineering Support), Dan Shropshire (FOT Engineering Support), Paul Viens (FOT Engineering Support), Ken Gage (FOT Mission Planning Support), Gordon Dressler (Propulsion Factory Support), Bernard Jackson (Propulsion Factory Support), Nancy Mayer (Propulsion Factory Support), Edward Chin (Factory Thermal Support), Perry Knollenberg (Factory Thermal Support), and Craig Cockerham (Factory Thermal Support).

Reference

[1] Chang, E. T., Poston, T. M., and Gokcen, N. A., "Density, Compressibility, Thermal Conductivity, and Mechanical Movement of Solid Hydrazine," Aerospace Corp. Rept. No. SAMSO-TR-77-31, El Segundo, CA, Oct. 1976.

Chapter 4

Validation of the Second ESA Deep Space Station Communications System

S. Casale,* F. Ardita,† and C. Cicero†
University of Catania, Catania, Italy
and
G. Buscemi‡
European Space Ageny, Darmstadt, Germany

I. Introduction

IN THE last few years ESA has initiated the procurement of a Deep Space Network to increase its capability to support deep space planetary exploration missions. The New Norcia Station, the first ESA deep space station, is located 140 km north of Perth (Western Australia). It hosts a 35-m-diam antenna with transmission and reception in both S- and X-band and provides daily support to Mars Express, Rosetta,§ and Venus Express. A second deep space antenna, similar to the one located in New Norcia but with several enhancements in performance and technology, has been operational since 2005 in Cebreros (province of Avila, Spain, about 70 km northwest of Madrid), and it is the prime tracking station for the Venus Express satellite.

As well as working on its own independent projects, ESA regularly cooperates with other public or private organizations such as universities and space agencies in Italy, the United States, Russia, Canada, Japan, and China.

In 2004, the European Space Operations Centre (ESOC) and the Dipartimento di Ingegneria Informatica e delle Telecomunicazioni (DIIT) of the University of Catania started to cooperate on drawing up and implementing a plan to test and validate the information and communications technology (ICT) system of the Cebreros station [1].

---

*Full Professor, Dipartimento di Ingegneria Informatica edelle Telecomunicazioni.

†Telecommunications Engineer, Dipartimento di Ingegneria Informatica e delle Telecomunicazioni.

‡Network Engineer, European Space Operations Centre, Directorate of Operations and Infrastructure, Operational Network Communications.

§Comet Wirtanen Rendez-vous Mission (Space Project).

ESOC, the satellite control center for the European Space Agency, provides all of the necessary ground facilities for missions in preparation and in flight according to the ESOC mission model. The Directorate of Operations and Infrastructure, Operational Network Communications (OPS-ONC) section is responsible for the operational ground communications facility providing support for 1) ESA missions in preparation and in flight, 2) third party missions, partner organizations [Center National d'Etudes Spatiales (CNES), Deutsche Forchungs-und Versuchsanstalt Für Luft-und Raumfahrt (DLR), Japan Aerospace Exploration Agency (JAXA) and NASA], 3) local area networks (LANs) at ESOC, Villafranca, and Redu, and 4) operational LANs and wide area networks (WANs).

The DIIT of the University of Catania has a staff of over 40 ranging from academics to researchers and Ph.D. students, who share and provide intellectual resources and knowledge for degree courses in computer science, telecommunications, and electronic engineering. It also organizes Ph.D. and master's courses. With regard to scientific research, the department is actively involved in several ICT projects, both national and European, and collaborates with other private and public organizations, such as ESA.

The principal research activities carried out in the telecommunications area by the DIIT can be summarized as follows: 1) distributed multimedia applications, 2) next generation Internet, 3) wireless networks, 4) multimedia traffic characterization and modeling, 5) transport protocols and resource management for mobile satellite networks, 6) wireless and satellite Internet protocol (IP) networks, 7) audio- and video-coding, 8) voice recognition in noisy environments, and 9) variable bit rate (VBR) speech coders for adaptive voice over IP (VoIP) applications.

The DIIT is a member of Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), a non-profit-making consortium whose main purpose is to promote research activity and provide networking support for specific projects in the telecommunications area.

Finally, the DIIT is also a member of Gruppo Nazionale delle Telecomunicazioni e Teoria dell'Informazione (GTTI), a national group embracing universities, public and private research institutions, and industries, with the aim of coordinating teaching and research activities in telecommunications at a national level.

II. Cebreros, the Second ESA Deep Space Station: An Opportunity for Cooperation

Being aware of how significant the international exchange of knowledge is for university students, the University of Catania has always encouraged and supported training programs for students both as part of their degree courses and as an opportunity to acquire personal and professional experience. A traineeship represents their first approach to the professional world, filling the gap that commonly exists between theory and practice.

A valuable opportunity for university students interested in space science and related areas is offered by the European Space Agency. ESA's internship program gives them the opportunity to work in one of ESA's establishments and gain important experience in an exciting international and multicultural environment.

The program helps students to meet the university requirement to carry out an internship as part of their degree course, and promotes their academic and professional development by placing them under the guidance of experienced professionals.

This kind of internship lasts for at least three and sometimes six to nine months and is addressed to young nationals from ESA member states in their last two years of study in the fields of mathematics, engineering, physics, science and information technology, as well as law, economics, etc. (ESA member states are Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxembourg, the Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, and the United Kingdom.)

Internships are, in fact, available in a wide variety of disciplines and can be divided into the following areas:

1) Technical, which includes space science, astronomy, mechanical and electrical engineering, telecommunications, earth observation, aerospace engineering, navigation, microgravity research, the exploitation of spacecraft in orbit, satellite tracking exploitation, the processing and distribution of data from remote sensing satellites, information technology, etc.

2) Non-technical, which includes law, contract law, international affairs, communications/public relations, finance, human resources, etc.

The long-standing professional relationship between ESOC engineers/researchers and academics from the University of Catania paved the way for greater collaboration in carrying out testing activities on the communications system for the latest generation ESA deep space ground station.

The need to validate the ICT system of the Cebreros station was considered a challenging opportunity for university students, giving them an opportunity to contribute and give support to the Cebreros project and put into practice the knowledge and experience they had acquired up to then on their academic courses. The collaboration established between the University of Catania and ESOC with a series of technical traineeships was significant for both the students, as an excellent opportunity to grow and work in a multicultural, competitive, stimulating environment, and for the agency, as a way of maximizing capabilities and resources. The collaboration lasted from September 2004 to September 2005, and it involved four final-year students in different phases of the 'Cebreros Communications System Test and Validation Plan.

III. Work Approach

The overall test plan for the Cebreros ICT system identifies and addresses a set of tests and procedures to validate the operational and non-operational cebreros communications system.

The first step, called *In-House Configuration and Functional Test*, identifies a set of tests and procedures that validate the configuration and functionality of Cebreros communications system network devices. The tests were performed in the ESOC laboratories after all of the network equipment had been installed and configured. The purpose of these tests was verification of the network design and evaluation of the main performance characteristics.

The second step, called *NDS-Factory Acceptance Test (FAT) and Regression Test (FAR)*, was performed at the NDSatcom laboratories, Friedrichshafen, Germany, which is the prime ESA contractor for the definitive integration of the Cebreros back-end system (including the communications system). During this phase, all of the Cebreros network equipment (routers and switches) tested in the ESOC labs was moved to the NDSatcom premises, and it was integrated and tested together with other communications systems (i.e., intercom system, audio and video facilities) in the real racks of the final configuration to be installed in Cebreros. The purpose of these tests was verification of the network functions to allow the correct data flow between the remote station and ESOC before on-site installation and operational activity.

The third and last step, the *On Site Acceptance Test*, was performed on site, at the Cebreros station in Spain. During this phase, all of the communications equipment, already mounted on the real racks of the final configuration, was rechecked and tested, and the results obtained were compared with those from the previous two sets of tests. This test phase represents the final step to validate the whole ICT system under nominal working conditions.

During each phase of the test plan, the trainees were actively involved in the test activities, making a valid contribution to the preparation, execution, and documentation of the test plan.

IV. Cebreros: Communications System Overview

The Cebreros station (Fig. 4) is the latest generation of ESA deep space antenna undergoing the deployment of a transmission control protocol/internet protocol (TCP/IP) architecture according to the ESA OPSNET strategy of migration to an infrastructure that can support all of its user services by relying solely on the TCP/IP protocol suite. This means that it is possible to support data, voice, and video applications by using a single IP platform.

The modernized physical infrastructure in the Cebreros ground stations is based on standards and best practices for structured hierarchical building cabling systems. One physical access point is capable of supporting all applications. The cabling towards end-user systems is gigabit capable and hence suitable for the foreseeable future. The LAN backbones are based on fiber optics with gigabit Ethernet interfaces. For critical services, two independent sets of devices are deployed end-to-end for redundancy (referred to as independent “chains”).

The overall network architecture is a classical hierarchical three-layer model. The layers are 1) core, 2) distribution, and 3) access.

It is this architecture, and its implementation by modular equipment, that makes the network highly flexible for adding sites or/and user systems.

The access layer provides the first point of access to the network for a connected system. The distribution layer handles the switching of data streams, security, and the grouping of user systems into different logical entities, virtual LANs (VLANs). The distribution layer also aggregates links based on the same groups, and implements routing and security within the campus. The core layer of the network is designed to handle routing between distant sites, including re-routing in case of outages of WAN links. The core functionality resides mainly in the core routers at the control center, but a few tasks are shared with the routers at the ground stations.

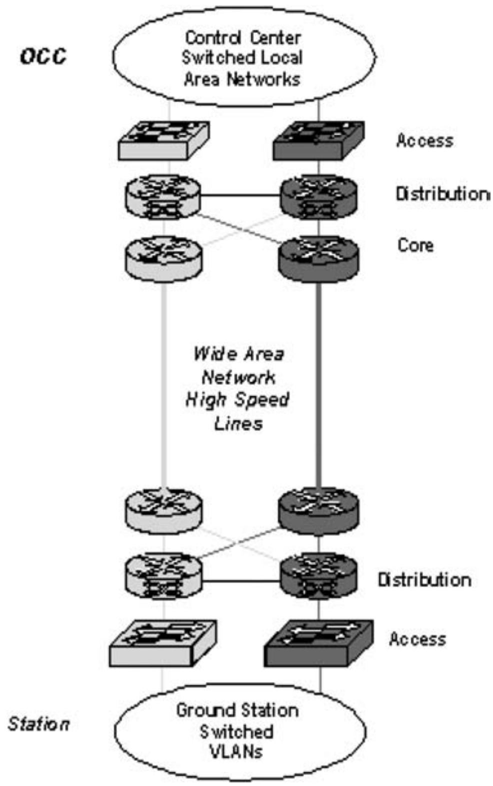


Fig. 1 Cebreros: network architecture.

Figure 1 illustrates the principle, shown for a link between the operations control center (OCC) and the Cebreros ground station, showing the redundancy of the two "chains." As can be seen, there always remains at least one path between user systems in the OCC and those in the station even if a leased line or in-between communications equipment fails.

The building of ESA deep space stations, each as an existing ground station, has been complemented with an enhancement of the operational support network (OPSNET) WAN, topology in Australia and Spain. Traditionally, OPSNET was a star network with ESOC as the hub, where each outstation connects to the control center via two diversely routed international leased lines as a redundancy pair. With the advent of the deep space stations, the topology of dual international links per single outstation could be replaced by a ring topology [2]. Figure 2 illustrates the ESOC–Villafranca–Cebreros ring.

An analogous layout exists for the ESOC–Perth–New Norcia ring. The physical infrastructure conditions end to end must of course ensure that none of these links has an element of failure in common with another link. The three links then form a ring in which any site can still communicate with any other site even if one link fails. The economic benefits are substantial. Instead of four international lines in

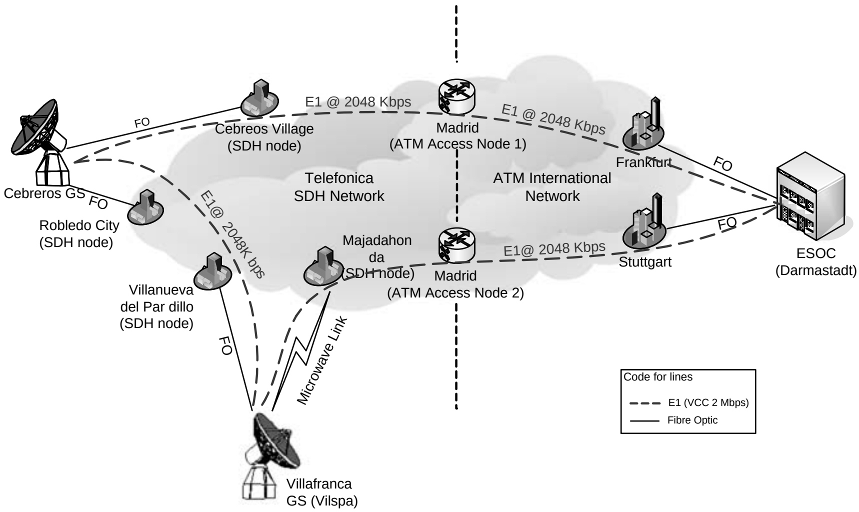


Fig. 2 WAN connection topology.

the same foreign country, two can fulfill the needs. The capacity per line in the ring has to be larger than per line in a star, but this is no disadvantage, as the ratio of price increase per capacity increase is strongly digressive. The redesign in fact paved the way to using lines of 2 megabit/s as standard building blocks for the rings. This type of link had in past years become the market offer with the best price/capacity ratio for the type of trunks required for the ESA OPSNET, and is also deployed elsewhere.

At the Cebreros ground station, an 18-switch/router device has been deployed offering approximately 1000 ports in the access layer.

With this capacity, Cebreros has the largest ICT infrastructure of all ESA ground stations. The high number of ports means that, in addition to "traditional" ESA ground station data and voice services, the Cebreros ICT infrastructure supports much more. All antenna front- and back-end equipment monitoring and control, previously still based on dedicated bus structures, is now supported over IP. Further, IP telephony is deployed; LAN ports provide both the channel and the electrical power for the IP phones. Audio- and video-conferencing, and video-distribution are also served, as are building facility management functions. With this thorough LAN technology concept, the only media that are needed between different buildings are optical fibers. This has great benefits for electrical grounding conditions and lightning protection.

The Cebreros Communications (COMMS) system can be split into four logical segments:

1) Operational LAN. This covers all parts of the station that are necessary for remote operation of the station. It is in charge of monitoring and controlling the equipment, data transmission to or data retrieval from the control center, and maintenance operations.

2) Office LAN and telephone. The Office LAN has the same logical design as the Operational LAN but is physically and logically separated from it, with a separate set of equipment to prevent any interference between the two LAN systems. The Office LAN includes the following equipment and services: communications equipment (Non-operational equipment), IP phones, digital cordless (DECT) phones, office computers, printers, faxes, Internet access, wireless connection to the office LAN (IEEE 802.11b).*

3) Intercom system. The Cebreros voice conference system (VCS-Intercom System) is a voice communication system connecting the operational building main equipment room (MER), the power building (PWR), and the antenna equipment room (AER) building. It also interfaces with the control center, ESOC, and the station at Villafranca, VILSPA.† It allows voice connections among all of these sites for critical and maintenance operations.

4) Audio- and video-conferencing system. The station is also equipped with an audio- and video-conferencing system connected to the private branch exchange (PBX) through which the audio traffic is provided via integrated services digital network (ISDN) lines or via the spare capacity of the leased lines (PBX connected to the operational router as shown in Fig. 3).

Key winning features of the new Cebreros ground communications system are the following:

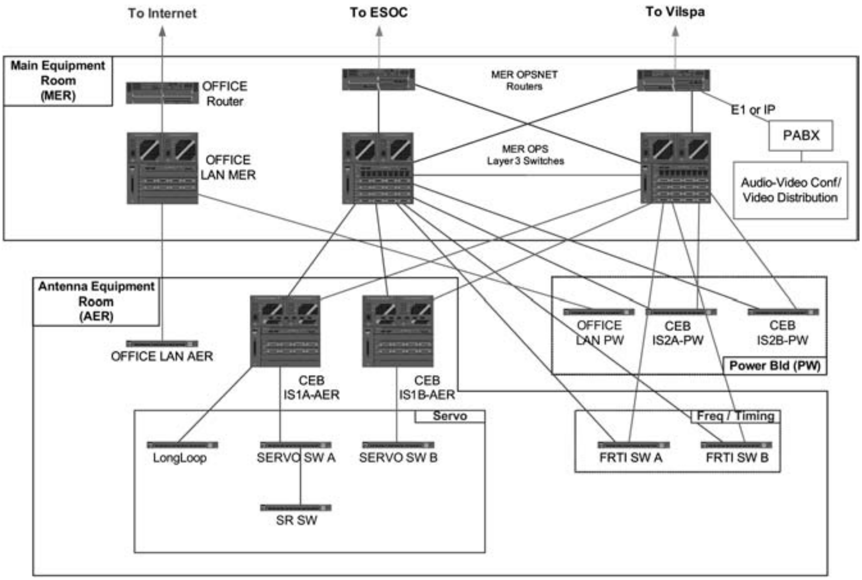


Fig. 3 Cebreros communications system, LAN diagram.

*IEEE 802.11b is a standard that specifies carrier sense media access control and physical layer specifications for 5.5-Mbps and 11-Mbps wireless LANs operating in the 2.4-GHz frequency band.

†Villafranca Space Antenna.

1) Capability for high performance at low procurement and running cost.
2) High flexibility and adaptability at low maintenance and change management cost.
3) Availability > = 99.95% on average per month.
4) Redundancy for critical devices, highly automated fail-overs.
5) Powerful quality of service and prioritization features.
6) Critical data get guaranteed capacity even in adverse conditions.
7) Non-critical data get capacity on best-effort basis.
8) Scalability/flexibility of network architecture in principle and of station installations.
9) Dedicated logical LAN segments per function/purpose (such as telemetry telecommand, monitoring and control, intercom, office automation, telephony, video-conferencing, audio-conferencing, Internet access).
10) Centralized network management in the control center, local systems management in stations available if needed in contingency scenarios.
11) Longevity of installations and equipment.
12) Coexistence with the X.25 network, with X.25-based mission operations.
13) Low procurement cost, low running cost (devices, telecom services, maintenance and operations and sustaining engineering services).

The Cebreros station has a hybrid topology, which is a combination of a partially meshed network and a flat network. This type of topology allows fault-redundant, automatic recovery and load balancing when a link or a network element fails. As shown in Fig. 3, all of the network devices are distributed between different locations, the MER, AER, and PWR buildings. The fiber optic backbones run gigabit Ethernet interconnections between buildings.

User systems are interconnected over a redundant mesh of fast- and gigabit-Ethernet routed/isolated IP connections. This infrastructure, known as L3-mesh, increases resiliency to failures and controls traffic patterns.

Between the OCC and the station, the dynamic routing protocol, the enhanced interior gateway routing protocol (EIGRP) is configured to handle re-routing in the event of failures.

For throughput guarantee and management of congestion on the WAN, a sophisticated quality of service (QOS) system called Class Based Weighted Fair Queuing (CBWFQ) has been deployed in the distribution layer at the ground station.

To set up CBWFQ, exhaustive traffic matrices have to be established for all transfers to be supported over the WAN links. Table 1 is an example of such a matrix. For each application or service, an allocation is made for the minimum throughput that is guaranteed, even if all services are active at the same time and competing with each other.

The capacity for a particular service is, however, provided only when that service is indeed active. If not, the spare capacity may be used by other services. Each service will thus get a much higher throughput if the momentary overall load of the trunk allows it. An exception in terms of guarantee is the interconnection between the telephone systems (PBXs). It can use this peak capacity only if the accumulated load for the other services does not prevent it.

All communications installations in the stations are monitored and controlled 24 hours a day on each day of the year by the operations control center at ESOC.

Table 1 Example of QOS policy for the ESOC-CEB link

Cebreros ↔ ESOC policy	Cebreros → ESOC	ESOC → Cebreros
Traffic type	BW, kbps	BW, kbps
IP Basic NMS Functions (SNMP, etc.)	20	20
Intercom Voice Over IP ESOC ↔ Cebreros	24	24
Cebreros Station Computer	20	20
Cebreros Station Automated Test Tool	10	10
Cebreros GPS Data	10	10
Cebreros Intermediate Frequency Modem System	20	20
Cebreros → ESOC Telemetry (Mission A, on-line)	258	
Cebreros → ESOC Telemetry (Mission B, off-line)	128	
ESOC → Cebreros Telecommand		10
ESOC → Cebreros TM Acknowledgment		10
Cebreros ↔ ESOC TC Acknowledgment	10	
Cebreros ↔ ESOC Voice PBX (ceiling value, best effort, no guarantee)	600	600
Cebreros ↔ ESOC Building Management (future)	64	64
Cebreros ↔ ESOC CCTV (future)	128	128
Cebreros ↔ ESOC Access Control System (future)	64	64
Cebreros Subtotal	1356	980
Vilspa ↔ ESOC Backup Policy	400	400
Cebreros ↔Vilspa Selective Backup Policy	44	44
Trunk Total	1800	1424

A customized network management system (NMS) tool (a combination of HP OpenView and CiscoWorks products), is available at the station and at ESOC.

The station network equipment responds to standard simple network management protocol (SNMP) polls and generates SNMP alarms upon encountering configuration or status changes.

The nominal channel for SNMP transfers is in-band capacity, i.e., capacity provided by and via the managed objects themselves.

However, out-of-band access is also possible, allowing remote access to be maintained if nominal access paths are unavailable. Asynchronous access to a router console port via the packet network, if needed in combination with on-demand ISDN for leased lines backup, is a typical contingency scenario.

V. Test Description

In light of the design adopted, particular attention was paid to setting up the Cebreros Test and Validation Plan and organizing the test procedures in such a way as to highlight the benefits of the ICT solution adopted for the station design.

As mentioned previously, different sets of tests were performed with the purpose of verifying all network functions that allow the correct data flow both locally

Fig. 4 Cebreros station, Spain.

and between the station and the OCC in ESOC. The following is a brief description of the three sets of tests performed.

A. In-House Configuration and Functional Tests

The main purpose of this set of tests was to configure the equipment in the ESA laboratory and to verify the design concepts regarding the network protocols, performance, and user services. The tests evaluated different WAN and LAN emergency scenarios, like the failure of equipment, infrastructure and links, including the tuning and stressing of specific protocol and network parameters such as the time to recovery of the EIGRP. The following steps were processed:

1) Nominal network simulation. All the Cebreros switches and routers were mounted on the lab racks, configured, and connected to a test bed simulating the real scenario like the ESOC-VILSPA-CEBREROS WAN triangulation (Fig. 2).

2) Network device fine tuning of the different interfaces, routing protocols, management protocols, etc.

3) Fault simulation and performance tests. This set of tests included WAN and LAN link/equipment failures, core mesh failures, and the convergence time of recovery protocols like the rapid spanning-tree tests. In particular, the aim of the tests was to calculate and verify the total network convergence after a failure, which was artificially created from time to time by physically disconnecting a cable or turning a device off. WAN link failures, core mesh failures, router and layer 3 switch failures included evaluation of the time to recovery of the EIGRP protocol and checking of the new routes, guaranteeing the triangulations were correct. Verification/evaluation of LAN convergence included tests for the failure of different trunks that originate undesirable loops, which was simulated by physically disconnecting the cables. The purpose was to evaluate the convergence time of the rapid spanning-tree protocol (RSTP) and verify the spanning-tree topology implemented in the LAN (see Table 2).

Table 2 Summary of convergence time collected

In-house configuration and functional tests				
Tests	Type	CEB-ESOC, s	CEB-VILSPA, s	VILSPA-ESOC, s
1	WAN link failures (EIGRP)	2	0	0
2		0	6	0
3		0	0	2
4		0	1	0
5	Core mesh failures (EIGRP)	2	0	——
6		2	0	——
7		2	2	——
8		0	2	——
9		3	0	0
10	Router failures (EIGRP)	0	5	0
11		0	0	2
12		0	4	0
11	Layer 3 switch failures (EIGRP)	0	2	2
13		0	0	0
14	LAN trunk failure (RSTP)	1	0	0
15		0	0	0
16		0	2	0
17		0	0	2

B. Factory Acceptance Test and Regression Test

The test phases for the Factory Acceptance Test (FAT) and the Factory Regression Test (FAR) at ND SatCom AG (NDS) in Friedrichshafen, Germany, are the following:

1) Visual inspection of the network integration performed according to the standard integration rules for ESA ground station communications systems.

2) Testing of the subsystems: intercom, video-conferencing, video-distribution.

3) Operational network tests under nominal and fault conditions: a) core mesh failures; b) verification/evaluation of LAN convergence and LAN performance measurement; verification of the correct data flow and performance (in terms of throughput) during a file transfer between two PCs connected to the Cebreros operational switches or two PCs connected to different places, ESOC and NDSatcom; and c) verification of the local and remote monitoring facility. A summary of the convergence time for core mesh and trunk link failures is shown in Table 3.

4) Office network tests. The whole office and telephone system was checked, verifying that the all of the equipment, like office PCs, IP phones, DECT phones, printers, Fax and IP PBX, were deployed and connected. Cebreros will be the first ESA ground station supporting IP phones. QOS therefore has to be configured on the switches to prevent interference with other traffic flows and to assure

Table 3 Summary of convergence time collected

NDS-Factory Acceptance Test (FAT) and Regression Test (FAR)			
Tests	Type	CEB-ESOC, s	RSTP test, s
1	Core mesh failures	2	——
2		1	——
3		0	0
4		0	1
5	LAN trunk failure	1	0
6		1	1

a good voice quality even during file transfer. This includes a) consolidation of the office switch configuration for IP telephone and office connections; and b) testing of the IP telephony in different scenarios and evaluation of performance with office data traffic.

C. On-Site Acceptance Test

In this final step, the complete system was rechecked and reevaluated with the real operational scenario and missions operations according to two the phases. First, the testing of the operational network included the following:

1) Re-check of the router and switch configurations.

2) Verification of the local and remote management facility, maintenance and operation (MO).

3) Verification and evaluation of WAN/LAN convergence in different fault conditions.

4) Measurement of WAN/LAN performance.

5) Test of the intercom system and voice loops.

Second, testing of the non-operational (office) network included the following:

1) Verification of office LAN and IP telephone system connections.

2) Testing of the audio and video facilities (audio, video-conferencing system, and video-distribution system).

After more than 2000 physical, logical, and performance tests, the ESA's second deep space station was transferred to the routine operations team with the tangible result of interconnecting system capable of connecting all space mission supporting systems, from the control center to stations, based on IP as the single data transmission protocol.

D. Test Results

The most salient results for cases of emulated failures are listed in Table 4. They are the maximum measured convergence times of an end-to-end connection, i.e., the maximum time it takes for a failed traffic flow to be active again after a failure.

This activity demonstrated that the results achieved in the real network were similar or in some cases better than those obtained during the initial phases,

Table 4 Summary of convergence time achieved

In-house configuration and functional tests			
WAN link failures (EIGRP max conv. time)	Maximum values achieved by physically disconnecting the serial cables connecting CEB-ESOC-VILSPA		
	CEB-ESOC	CEB-VILSPA	VILSPA-ESOC
	2 s	6 s	2 s
Core mesh failures (EIGRP conv. time)	Maximum value achieved by physically disconnecting the cables among the 2 routers and layer 3 switches making up the L3-mesh level. 3 s		
Router failures (EIGRP conv. time)	Maximum value achieved by only turning down the routers among the stations ESOC-VILSPA-CEBREROS (Fig. 2). 5 s		
Layer 3 switch failures (EIGRP conv. time)	Maximum value achieved by only turning down the two layer 3 switches connected to the routers making up L3-mesh level (Fig. 2). 2 s		
LAN trunk failures (RSTP conv. time)	Maximum value achieved by disconnecting the cables among the LAN switches only implementing the RSTP protocol 2 s		
NDS-factory acceptance test (FAT) and regression test (FAR)			
Core mesh failures (EIGRP conv. time)	Maximum value achieved:	2 s	
LAN trunk failures (RSTP conv. time)	Maximum value achieved:	1 s	
On-site acceptance test			
Core mesh failures (EIGRP conv. time)	Maximum value achieved:	less than 1 s	
Layer 3 switches failures (EIGRP conv. time)	Maximum value achieved:	less than 1 s	
LAN trunk failures (RSTP conv. time)	Maximum value achieved:	1 s	

considering that, for example, real links provide immediate live diagnostic (e.g., “carrier detect” signal down on the physical interface), and immediately apply the re-routing policy; a simulated failure, takes around 4 for re-routing algorithm calculation on the loss of a link.

VI. Benefits of the Collaboration

Since the cooperation started, we believe that all of the established targets have been reached from both a technical and a cooperation point of view. All of the activities carried out, together with the results achieved, validate the Cebreros communications system, highlighting the benefits of the design chosen and the

associated ICT solutions adopted. After an initial setting period during which they learned how to manage the network tools necessary for the tests, the trainees involved took an active part in the test activities, becoming part of a strongly motivated team and participating in activities that involved people and companies of different nationalities. The main role that this kind of ESA internship program has is “to inspire and motivate students to pursue careers in space.” ESA also gained the opportunity from this experience to exploit academic knowledge in an important project, thus improving some of its own technical results and reducing the costs.

For the University of Catania, on the one hand the experience represents a remarkable point of reference for further cooperation in European space activities, and on the other could serve as a stimulus to enhance education and research activities in the space field, promoting participation in other ESA programs.

In the last analysis, the trainees had the opportunity to explore the entire project regarding the ground communications segment for the deep space station and familiarize with the technical aspects of the different types of communications equipment like audio- and video-conferencing systems, switches, routers, and so on. They also had a good interaction with highly qualified ESA people working on the project, thus maximizing the technological achievements. Each trainee will benefit in the future from the ESA variety of cultures, different languages, and the highly motivated work environment. The cooperation represented a valuable high-level work experience that will be useful to them in their future professional careers.

VII. Conclusion

The tests were carried out punctually, respecting the planned scheduling. All of the results obtained in the last phase performed on-site are in accordance with (in some cases even better than) those filed in both the ESOC laboratory and NDSatcom, revealing that the system works properly under nominal working conditions. All of the work was collected in about 200 single tests and reported in an ESA document [3].

The whole activity and related documentation certainly constitutes a significant starting point for the development of a systematic test and validation plan before on-site installation and operational activity and, together with the ESA’s multiyear experience in the design and installation of ground stations, represents a valuable lesson to be taken into consideration for the design and implementation of new ground stations, in particular the communications system for the third ESA deep space station.

Further, thanks to the ESA internship contracts, the DIIT and the OPS-ONC departments had the opportunity to involve all of the trainees in same project continuously and for a long period of time. In this way, the students progressively improved their approach to the work thanks to previously gained experience.

This kind of approach can therefore be applied to future cooperation where it is necessary to have human resources available for long-term projects allowing the

sharing of knowledge, hands-on experience in important projects related to space activities, and cooperation in a multicultural and stimulating environment.

Acknowledgments

The authors thank the ESA HR for its support during the internship period, the colleagues in the Ground Station Systems Division, the Ground Facilities Operations Division, ND SatCom, the Vega IT GmbH communications engineering team, and the maintenance and operations teams at the control center and at each ground station.

References

[1] Casale, S., Buscemi, G., Ardita, F., and Cicero, G., "Cooperation Between ESA-ESOC and the University of Catania for Testing and Validation of the 2nd ESA Deep Space Station Communications System," *SpaceOps 2006 Conference*, 19–23 June 2006, Rome, Italy, pp. 1–6, AIAA Paper 2006-5874.

[2] Bertelsmeier, M., Buscemi, G., and Mamedov, F., "A Novel and Cost Effective Communications Platform for ESTRACK," *6th International Symposium on Reducing the Costs of Spacecraft Ground Systems and Operations*, ESA/ESOC, Darmstadt, Germany, June 2005.

[3] Buscemi, G., Mamedov, F., Bellistri, M., Cicero, C., and Ardita, F., "Cebreros Deep Space Ground Station – Communications System Test and Validation Plan," (DS2-ESC-PL-7000, DS2-CEB-COM-TP-1001-OPS-ONC), ESA/ESOC, Darmstadt, Germany, Sept. 2005.

[4] Bertelsmeier, M., Buscemi, G., Butkovic, M., Mamedov, F., and Matute, J., "The Evolution of ESA Ground Station Communications to Internet Protocol," *Seventh International Symposium on Space Mission Operations and Ground Data Systems*, http://www.aiaa.org/Spaceops2004Archive/papers/index_e.asp.

[5] Bertelsmeier, M., Buscemi G., and Mamedov, F., "Approach for Ground Stations Communications Within the ESTRACK Network of ESA," DASIA 2005.

This page intentionally left blank

II. Standards

This page intentionally left blank

Chapter 5

CCSDS Cislunar Communications Architecture

Keith L. Scott*
MITRE Corporation, McLean, Virginia
and
Christopher J. Krupiarz†
Johns Hopkins University, Applied Physics Laboratory, Laurel, Maryland

I. Introduction

SEVERAL space agencies are embarking on programs of human exploration of the moon and possibly other planets that will involve orders of magnitude more elements than have been previously deployed. Elements both within and between missions will need to communicate with each other to perform command and control as well as data return. Multi-hop communications will be required for configurations that do not support direct line-of-sight communications. This chapter describes a networked communications architecture developed by the Consultative Committee for Space Data Systems (CCSDS) that provides a framework for interoperable communications among space elements, ground stations, and terrestrial users. This cislunar communications architecture allows for a combination of CCSDS packet telemetry and telecommand, Internet protocols (IP), and delay/disruption tolerant networking (DTN) overlays to provide a range of communications mechanisms that can be tailored to the particular environment and mission.

Mission designers have traditionally taken a point-to-point approach to spacecraft command, control, and communication. Except in unique circumstances where missions consisted of multiple spacecraft, commands, telemetry, and voice communication flowed directly from an Earth-based ground station to the spaceborne asset. While this stovepipe solution is suitable for scenarios

---

*Principal Engineer, Network and Communications Engineering Department.

†Senior Professional Staff Member, Space Department, Embedded Applications Group. Member AIAA.

where there are a limited number of spacecraft acting individually, it becomes increasingly difficult when dealing with spacecraft that need to communicate among each other, and as the number of spacecraft operating at any one time increases.

Such an occurrence is on the horizon with regard to cislunar communication. The cislunar environment, defined as the region between Earth and the moon, will see a many-fold increase in exploration activity in the next decades. NASA has initiated a program called the Vision for Space Exploration (VSE) [1], which is targeting a human return to the moon; the European Space Agency (ESA) is developing the Aurora program, which will explore the surface of Mars; and countries ranging from India to Japan have constructed or are planning to send spacecraft to the moon.

To alleviate the amount of infrastructure necessary to facilitate these missions and to reduce the costs of operating mission unique systems, CCSDS created the Cislunar Working Group to develop a comprehensive communication architecture for the Earth/moon system that will meet the operating and interoperability needs of these diverse missions. The working group has developed an architecture that can be deployed incrementally, using current technologies and protocols where it is feasible, hybrid solutions where it is possible, and new technology where it is necessary. The goal is to reduce risk and costs, and to increase the amount of scientific returns from these missions.

II. Goals and Scope

This chapter describes an internetworking communications architecture for the cislunar mission domain. It does not prescribe communications architecture in general, such as relay satellite configurations, but it must make assumptions about the possible range of options for broader configurations, so that the internetworking functions described here will serve the wide range of missions (including both human and robotic) over the range of communications topologies (with relays, without relays, etc.) that could be deployed in cislunar space.

A major boundary of the cislunar mission domain is described in terms of communications delays. The architecture presented here is designed to function in the presence of round trip times (RTTs) that might include communications paths of lunar missions, such as relay through geosynchronous satellites, lunar orbiting satellites, and satellites in the vicinity of the Earth–moon Lagrangian point 2 (EML2). These geometries result in light-time delays of only 3–4 s round trip time (RTT). The “next” missions of interest are those at the sun-Earth Lagrangian point 2 (SEL2) with RTTs in the neighborhood of 10 s. Beyond that, the next expected mission set comprising interplanetary missions would incur round-trip light times on the order of 4–40 min, such as those to Mars or Venus. Communications protocols that function well at lunar distances will likely be extendable to SEL2 missions with 10-s RTTs. Further study would be necessary to determine if the proposed architecture could fully function at interplanetary distances. Thus the boundary of RTTs considered here is 10 s to accommodate missions in cislunar space or out to SEL2 distances. Figure 1 illustrates the basic areas where the communications architecture described here is intended to function. Mars’ two moons, Phobos and Deimos, lie within about 150 ms round-trip light time from the surface of Mars and so fall well within the 10-s region.

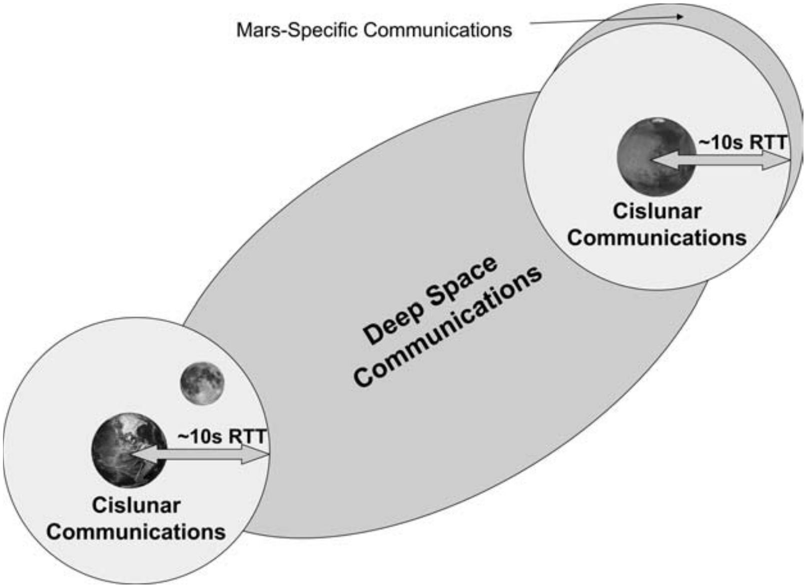


Fig. 1 Scope of cislunar communications architecture.

The concepts presented here should work equally well in any system with round-trip light times that are roughly equivalent to the Earth–moon system. Thus the arguments presented should hold for the immediate "local" vicinity of most planetary bodies in the solar system, and in particular should hold for communications between Mars' orbit and the planet's surface.

III. Communications Requirements

In many ways, the communications requirements for a space-based network are similar to those of a terrestrial environment. The network should allow for varying types of data to flow across the system, it should be scalable from a few spacecraft to many spacecraft or many computers running on individual spacecraft, and it should be interoperable across spacecraft, organizations, and nations. This section describes a number of desired aspects of the target architecture.

A. Flexibility

Voice, video, and data in the form of at least file transfers, electronic mail, and possibly instant messaging will be applications future network architectures will be required to support. Each of these has different requirements for service in terms of latency, jitter, and correctness of the data. In addition, any number of applications with applicability to the space domain, with yet different service requirements, may be developed over the next few decades.

Historically, space communications systems have carefully managed data volumes and data paths with well-defined point-to-point links, sometimes tying particular applications to specific physical resources. This tight coupling of applications to physical communications provides a high degree of safety and control, but makes it difficult to add new applications or new types of physical connectivity. It can make it difficult for multiple applications to efficiently share physical resources.

This argues for an isolation function separating applications from the underlying communications resources. Ideally, this isolation function should provide a powerful upper-layer (service) interface to applications, efficient multiplexing of multiple applications onto multiple physical resources, and the ability to arbitrate among competing application demands. To function in the full range of envisioned communications environments, the isolation layer needs to be able to provide at least some level of service over unidirectional data links, function across a wide range of delays, and accommodate situations where end-to-end connectivity is not always present.

B. Scalability

If one were to attempt to engineer custom interfaces between each pair of communicating elements, and then to manage multi-hop data flows through the resulting infrastructure, the complexity would grow at least as the square of the number of elements. This would quickly become unmanageable after just a few spacecraft. The system also needs to be scalable with the total number of endpoints and applications, not just the total number of data links. If the exploration efforts are successful in exploring and exploiting resources on the moon, future lunar bases might contain local area networks with tens or hundreds of computers, each running multiple applications. Also, exploration vehicles may contain dozens of unique computer operated devices that may require updates or operations across a network.

C. Interoperability

There are two distinct aspects of interoperability in the cislunar environment: interoperability among deployed elements in space and interoperability with terrestrial data communications. Both types of interoperability will be needed. Interoperability among elements will be required to perform docking maneuvers and to exchange science information among space-based elements. The scalability argument argues for a single interoperability standard to reduce the sheer number of interfaces between elements. While there will be cases where both endpoints of communication will be space based, the majority of communications, at least for initial missions, will involve both space-based and terrestrial endpoints. Thus interoperability between space-based elements and terrestrial ones is desirable. Finally, with respect to both space and terrestrial components, legacy systems will continue to be in existence for some time to come. Interoperability with existing systems will ease the transition process as the new architecture is being phased in and will provide increased opportunities for communication.

D. Cislunar Requirement Summary

In summary, the communication requirements for a cislunar communications system are the following:

1) Ability to support a wide range of applications with different requirements, including voice, video, and data.

2) Ability to operate effectively over a range of link and end-to-end delays.

3) Ability to function in the presence of simplex links and data paths.

4) Ability to efficiently support configurations from a few end systems to hundreds or thousands of end systems.

5) Ability to interoperate with legacy endpoints and communications.

IV. Networked Communications Architecture

A multi-hop, routed, packet-switched communications infrastructure can efficiently support the communications requirements of the previous section. This section briefly describes the terms multi-hop, routed, and packet-switched as they are used here, and then describes an architecture based on IP as a common end-to-end network layer packet. In the context of the previous requirements, IP provides the isolation function between applications and physical communications resources.

A. Multi-Hop

Communications paths will typically be several hops long, allowing different data links to be used in different environments. For instance, a typical data path might use Ethernet and SONET on Earth, space link protocols between the Earth and the moon, and Ethernet (wired or wireless) inside a lunar base. This contrasts with most current space missions, which are typically a single logical data link layer hop from the control center to the spacecraft. Note that in this case a Space Link Extensions (SLE) [2] tunnel from the control center to the ground station does not count as a separate hop, since SLE tunnels the terrestrial endpoint of the space link from the ground station through the Internet so that the space data link protocol is terminated at the control center.

B. Packet-Switched

Packet-switched communications allow for efficient multiplexing of many data sources onto a single physical layer link. This will be important in the cislunar environment because it will allow efficient sharing of data links among multiple users. Current space missions typically use CCSDS packets as their application-layer data units, and multiplex many different applications onto a single physical channel. The difference between this and a full packet switched network is that in the network case, packets from different applications from several different spacecraft may all be multiplexed onto a single downlink. While this is possible using the current suite of CCSDS protocols, there are limitations, including the lack of both source and destination address information in packets, standardized quality of service marking mechanisms, and standard automated forwarding procedures.

C. Routed

Intermediate relay points may want to make decisions on where to forward data based on per-packet information—not all data are going to be from source A going to destination B or even going in the same direction; think of an astronaut on the surface of the moon using a relay orbiter for beyond-line-of-sight communications with a both lunar base and engineers on Earth.

D. Cislunar Communications Architecture

The basic architecture described here is illustrated in Fig. 2. The rationale for standardizing a networked architecture rather than simply standardizing a set of data link layers is that inclusion of a network layer provides additional services that greatly reduce the complexity of the overall system while adding very little overhead. For example, a network layer provides a global addressing scheme. Without global addressing, an application would only be able to communicate directly with peer applications sitting on the same layer-2 (data link) segment.

Method, multi-hop data transfers would have to be managed hop-by-hop by the applications themselves, with potentially separate relay applications for each end-to-end application. This is the method employed by the Mars Exploration Rovers when transmitting data via the orbiters. The rover-to-orbiter links use the CCSDS Proximity-1 protocol, while the long-haul links to the Earth use CCSDS TM/TC. Special forwarding software on the orbiters is responsible for accepting telemetry from the rovers and forwarding that data to Earth.

IP provides standardized global addressing, along with multi-hop packet forwarding and routing capabilities, plus the ability to use quality of service markings that can be used to prioritize data. These allow applications to focus on the data they want to send without having to worry about the details of each forwarding hop. Destinations that may be multiple hops away are as easily accessible as those that are link-local. Standard transport layers such as Transmission Control Protocol (TCP) [3], Space Communications Protocol Specification – Transport Protocol (SCPS-TP) [4], and Stream Control Transmission Protocol (SCTP) [5] provide reliable stream and datagram services with congestion control, relieving applications from having to implement these features. The result is that an application using a reliable transport layer need only worry about its application-layer data; reliability, duplicate

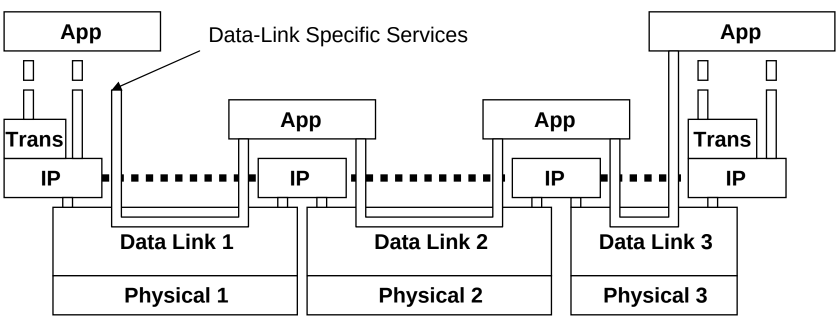


Fig. 2 Data paths in the cislunar architecture.

suppression, and the mechanics of moving data to and from the peer application anywhere on the network are all handled transparently to the application.

A disadvantage of moving to networked communications is that it is harder to leverage link specific services, since those services may not be available end-to-end. For example, the CCSDS Advanced Orbiting Systems [6] (AOS) service provides (among other things) an isochronous insert service for transmitting small fixed-length data units originally designed to support fixed-rate digitized audio. There is no standard mechanism to transfer insert service data from one AOS link to another, much less to remove it from an AOS link and forward it over an Ethernet or SONET link. Thus for an application to take advantage of this link-specific service, it would have to transmit directly to a peer application at the other end of the link. The networked architecture described here does not prohibit this kind of direct access to data link services; however, its use by "standard" applications is discouraged.

E. IP Packet Format

Most current space communications systems use the CCSDS packet as the common network-layer encapsulation format, which has greatly simplified operations since its introduction in the early nineties. While CCSDS packets could be used as the basis of a multi-hop routed approach, there are some disadvantages. These include the lack of both source and destination addresses in the packets, lack of a suitable quality of service (QoS) field in the packets, lack of a multi-hop routing protocol, and lack of wide-scale terrestrial support for forwarding CCSDS packets. While these drawbacks argue for a more capable networking layer, the experience with using CCSDS packets vs custom-built telemetry formats argues strongly for maintaining a common network layer packet format for simplicity, interoperability, and economies of scale in software/hardware development.

The IP developed by the Internet Engineering Task Force (IETF) is a layer-3 (network) protocol that is suitable for space communications. IP packets contain source and destination addresses, as well as fields for QoS marking of packets. Figure 3 shows the header format for IP version 4 packets.

There are currently two widely deployed versions of IP (IPv4 [7] and IPv6 [8]). The main differences between the two lie in the number of end systems they can address, and whether support for services like autoconfiguration, security, and mobility are mandatory or optional. While the terrestrial world is definitely

<table>
<tr><td>Ver</td><td>Header length</td><td>Type of Service</td><td colspan="2">Total length</td></tr>
<tr><td colspan="3">IP Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source IP Address</td></tr>
<tr><td colspan="5">Destination IP Address</td></tr>
</table>

Fig. 3 IP version 4 header.

moving to IPv6, it is not clear how quickly that transition will take place. In the meantime, products based on IPv4 tend to be more mature than their IPv6 counterparts, and network engineers are more familiar with IPv4 features and configuration than with IPv6. Fortunately, systems can opt for a mix of IPv6 and IPv4, and there are a number of solutions for running different protocols in different parts of the network, including tunneling each protocol across the other and/or using gateways to translate between the two. In cases where the overhead of either version of IP is too great, network-layer gateways can be used to convert between IP-based and the more bit-efficient CCSDS Space Communication Network Specification-Network Protocol (SCPS-NP)-based networks.

F. Flexibility, Scalability, and Interoperability of IP

The internet protocol itself is mostly a format for identifying the source and destination of data packets, along with information about which upper-layer transport protocol is being used to carry that data and space to hold QOS information. Because it isolates upper-layer protocols from the individual data links, IP provides a strong buffer between applications and underlying communication hardware. IP does not have any sort of closed-loop control between data source and destination, and hence can be used to carry information across simplex links. The main disadvantage of IP in a cislunar context is that all known IP implementations are intolerant of disconnection. That is, if there is not a continuous end-to-end path from the source to the destination, IP packets will be dropped at the point where they cannot be forwarded. This deficiency is addressed by current work in delay/disruption tolerant networking [9] discussed next.

IPv4 allows for the use of private address spaces that are 224 bits, or 16,777,216 hosts. This would seem to be enough to support cislunar operations for a long time to come. Another aspect of IP's scalability is in the way it supports upper-layer protocols through the protocol ID field. While TCP and UDP are by far the most popular transport protocols in the terrestrial Internet, there is ample room to define new transport protocols for cislunar operation should they be required. Multiplexing of applications over particular transport layers is handled by the transport layers, via 16-bit port numbers in TCP and UDP.

Standardizing on IP or any network protocol for cislunar communications will provide interoperability between deployed elements. The advantage of IP over other network protocols comes from the large-scale deployment of IP on Earth, particularly the Internet and many private networks. Choosing IP for cislunar leverages the huge array of applications designed for use in the Internet ranging from diagnostic and network management applications to electronic mail, web browsing, and instant messaging.

G. Automated Data Forwarding

The next step in advancing the efficiency of the infrastructure is to allow for automated forwarding of data packets. Automated forwarding will allow IP packets to flow from source to destination across a number of different data links without humans having to command each individual data moving operation. This should decrease operations costs, increase reliability, and reduce the latency of

data delivery. This will become increasingly important as the number of deployed elements grows, and will allow mission operations to focus more on data collection and analysis rather than the mechanics of data transport.

While much of the Internet uses routing protocols to automatically and dynamically update the forwarding tables of intermediate routers, this need not be the case for space missions. While we believe that dynamic routing may one day be used, either because it is needed to deal with the number of systems or simply to reduce operations costs, initial operations may choose to manage all forwarding tables via management interfaces. What this amounts to is telling each router, for each destination IP address, where packets for that destination address should be sent next (the next hop). This will allow mission operations personnel to maintain absolute control over the forwarding process while still not having to manage each individual data transfer across each link. For early missions when the number of in-space relays is small, managing all of the routing tables by hand will not be overly burdensome to operations personnel.

The management of the data forwarding tables takes place at the network (IP) level, and is insensitive to data link and physical connectivity underneath. Specifically, bent-pipe relays are "invisible" to the IP forwarding and routing mechanisms, and may be intermixed in the system along with other types of links such as 802.11G, switched Ethernet, and frame relay.

H. Quality of Service

One of the issues in moving away from the current highly managed communication model is that if the network becomes congested, packets can be dropped. For certain classes of traffic, data loss could be catastrophic, leading to loss of life or spacecraft. IP provides a number of quality of service mechanisms, ranging from differentiated services (diffserv [10,11]) to integrated services [12] to traffic engineering [13] that can be used to influence which traffic is forwarded first and which traffic is dropped when congestion occurs.

1. Differentiated Services

Differentiated services allows for different treatment of packets based on a field in the IP header. Using the standard definitions, packets are grouped into four assured forwarding (AF) classes, with three drop precedences per class. An additional diffserv per hop behavior, expedited forwarding (EF), is relevant for voice or emergency traffic that needs a high probability of delivery. Packets marked for EF should be forwarded first and dropped last by routers. This means that high-priority voice traffic marked for EF delivery would not be dropped due to network congestion, and would be delivered with minimal latency.

2. Integrated Services

Integrated Services, often referred to by the name of the signaling protocol, the resource reservation protocol (RSVP), allows applications to interact with the network to reserve communications resources (usually bandwidth). Integrated

services can provide applications with latency and congestion characteristics comparable to when the application is operating alone on the network, regardless of the actual traffic load. Like packets marked for expedited forwarding in diffserv, packets conforming to an existing RSVP reservation should not be dropped and should only see queuing from other packets in the same reservation. Thus voice packets would not be starved for network bandwidth due to large file or video transfers, for example. Disadvantages of RSVP are that it requires end-to-end signaling to reserve network resources, and reservations are not guaranteed to hold if the network path changes.

3. Traffic Engineering

Traffic engineering involves forcing packets to take paths in the network that they might not otherwise take to spread the load among multiple parallel links, to reduce congestion in parts of the network, or to reserve bandwidth in parts of the network for high-priority traffic. For early missions where the routing tables of all nodes will likely be managed by hand from the ground, essentially all traffic will be engineered. As time goes on and dynamic routing protocols are adopted, traffic engineering may be used to route low priority traffic via longer or less reliable paths, leaving high-quality paths for real-time traffic such as voice and video.

I. Emergency Commanding

Emergency commanding of spacecraft in a networked environment poses a number of challenges. For spacecraft that are tumbling, for instance, an important metric is the number of bits required to send a basic command such as “save the spacecraft” (return the spacecraft to a known, stable condition). For a spacecraft that is on station but has a damaged receiver, transmitting a command such that it arrives at the damaged spacecraft with the highest power might be more important.

Traditionally, emergency commands have been handled by a hardware command decoder that is very close to the radio onboard the spacecraft. A particular bit string is included in a data link layer frame, and a correlater immediately following the demodulation process detects the bit string and acts on it. An advantage of this approach is that none of the rest of the spacecraft command and control system (including any network stack on board) needs to be functioning. Indeed, hardware commands to reboot the main spacecraft command and data handling system are usually considered in spacecraft design.

There are three basic mechanisms for emergency commanding supported by this architecture:

1) Emergency commanding via IP. This option relies on the standard communications mechanisms to send an emergency command to a particular spacecraft and to have it recognized. Such a command could be identified with a special transport protocol type, or could be included as the payload of a standard UDP packet. Using IP to route the command allows emergency commanding of elements that are not proximate to the element doing the commanding (multi-hop communications). Drawbacks of this approach are that it requires that either the full IP (and possibly UDP) headers be transmitted, or the header compression mechanisms at the receiver be working. In either case, a hardware detection mechanism could be

used to detect a “special” bit pattern and act on it accordingly so that the full networking stack would not have to be functional.

Measures would need to be taken to ensure that the “trigger” bit pattern did not show up as the payload of any data transmitted by the spacecraft. One way to achieve this would be to make use of the data link layer synchronization mechanisms to search for a trigger pattern only immediately following a data link layer synchronization marker. Using the data link frame marker to bound the search for a trigger bit sequence would rely on the ability to insert the target IP packet at the beginning of the data link layer frame, a capability that is not necessarily supported by current space data links.

If emergency commands are sent as UDP or TCP traffic, then there is the possibility of using standard security mechanisms to authenticate and verify them, at the cost of extra bits and the assumption that the software to perform the verification is functional.

2) Emergency commanding via link layer mechanisms. It may be possible to use current mechanisms to effect emergency commanding. CCSDS data links support a number of virtual channels (VCs) that are commonly used to segregate traffic of different types, including emergency commands. Variable-length data link layers such as the CCSDS telecommand (TC) standard are particularly good for this, since a short frame header can be followed by a VC identifier and then the emergency command itself. The main drawback of this approach is that it is not routable; emergency commands must be issued by a link-local neighbor. Also, fixed-length data link layer frames like CCSDS AOS tend to be long, impairing the ability to get a short command into a tumbling spacecraft.

3) Combination of IP and link-layer mechanisms. A third mechanism would be to use a combination of the previous mechanisms, using Internet-based protocols to get an emergency command to a special application resident at the penultimate IP hop, and to use link-layer mechanisms to get it to the destination. This has the advantage of being able to use link-specific mechanisms that may allow very short commands while still allowing those commands to traverse multiple hops in the network.

V. Cislunar Network Mission Application Example

A networked architecture for the cislunar environment will enable mission designers to move beyond the current “stove-piped” command and telemetry infrastructure typically seen in space missions. By providing a consistent underlying architecture, spacecraft can insert themselves into the network and receive data from not only their own mission or science control centers but also from other spacecraft or assets. This will enable new opportunities for cross collaboration between missions and a move toward event-based science.

An example of some of the features of this new architecture is shown in Fig. 4. In this scenario the focus is on space weather and its impact on astronauts and spacecraft as well as the distribution of solar weather events across the terrestrial network to scientists on Earth. The data flow in this example begins with observations by a solar sentinel at the sun-Earth L1 point. A spacecraft at this location serves as an early warning beacon for solar activities such as coronal mass ejections (CMEs) and solar energetic particle (SEP) events. The Advanced Composition

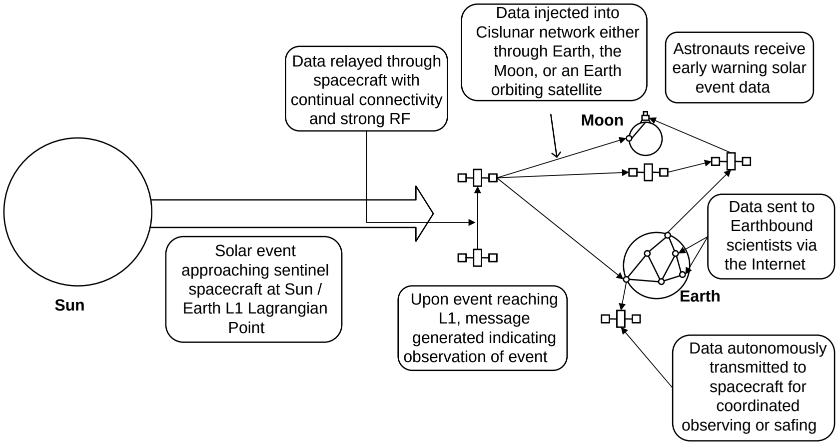


Fig. 4 Space weather distribution via the cislunar architecture.

Explorer (ACE) is an example of such a spacecraft, and it is currently positioned at the L1 point. Using the architecture described here, these events could be signaled by a publish-and-subscribe middleware application running on top of the underlying protocol stack. The Asynchronous Messaging Service (AMS) is an example of this type of middleware, and it is making its way through the CCSDS standards process. The AMS message would contain information pertinent to the event and would be transmitted back to Earth or, conceivably, to a lunar ground station as necessary either directly or through a relay satellite with a stronger R/F system located close to the sentry. At this point, it has entered the cislunar space network architecture and is available for distribution to entities subscribed to this message across the domain. This could include space weather researchers in a university setting, astronauts in need of early warning to abort EVA activities, spacecraft that would need to “batten down the hatches” to avoid possible disruption in their activities, or spaceborne assets who could activate instruments or perform other actions to ensure proper coverage of the event. The use of standardized protocols and messages makes it possible for new spacecraft or researchers to easily integrate themselves into this network to also receive this information. Additionally, a store-and-forward overlay would ensure that data were received by spacecraft that do not have continual real-time connectivity to the cislunar network.

VI. Issues with IP in Space Communications

While the bounds for the scope of the cislunar environment were chosen to make it possible to run standard Internet protocols, there are still issues unique to space communications that need to be addressed. Of particular concern to space mission designers is overhead on the highly constrained space links. Also, terrestrial networking generally makes the assumptions that round-trip times are low and that end-to-end paths exist. These assumptions do not necessarily hold, or

hold well, in the cislunar environment. This section addresses overhead concerns, the performance of TCP over large bandwidth*delay paths, and ways to deal with disconnection. Other issues include how to perform address resolution, management of name-to-address mapping (e.g., domain name service, DNS) services, choice of routing protocols, and security. All of these issues are addressed in more depth in the CCSDS cislunar Space Internetworking Architecture document being developed by the CCSDS cislunar working group.

A. IP Packet Overhead

Uncompressed, IPv4 headers require 20 bytes plus options, and IPv6 headers require 40 bytes plus any extension headers. Typical IP packet sizes tend to be either around 40 bytes (for TCP acknowledgments carrying no data), or 1500 bytes for full-sized TCP segments that traverse Ethernets. Because a typical TCP header is between 20 and 32 bytes depending on the options used, this means that uncompressed headers on a full-sized (1500-byte) data segment impose about a 3.5% overhead. For TCP acknowledgments that do not carry any data, the packet is essentially 100% overhead but necessary to support TCP's reliable data delivery.

A number of header compression mechanisms have been defined for both IPv4 and IPv6 that drastically reduce header overhead by compressing the IP and sometimes transport headers as well into a single 1- or 2-octet context identifier. This has the effect of reducing the effective size of an IP header to 1 or 2 bytes, rather than 20 (uncompressed IPv4) or 40 (uncompressed IPv6) bytes. For full-sized IP packets, this amounts to a header overhead of less than 1%. The benefits of using a network protocol, and IP in particular, including the ability to support multi-hop communications, automated data forwarding as described in the next section, and multicast greatly outweigh this minimal additional overhead.

B. TCP and Large Bandwidth*Delay Paths

The standard Internet protocol for reliable communications is TCP. TCP uses window-based flow and congestion control, meaning that the data sender can only send a given amount of data before it must stop and wait for a response (an acknowledgment) from the receiver. The amount of data the sender is allowed to have outstanding at any one time is the sender's window. The larger the product of the bottleneck bandwidth and the round-trip time of the path, the larger the sender's window has to be to allow it to transmit data continuously and to keep from having to leave the connection idle.

Another issue with running TCP in the cislunar environment is a result of the round-trip times alone. TCP's congestion control mechanism operates on the timescale of the round-trip time of the connection. When the round-trip time is large, TCP is very slow to react to changes in path bandwidth and also very slow to recover from any network congestion. While mechanisms exist to allow TCP to use large windows, large round-trip times remain a problem.

One solution that has seen widespread use in terrestrial networks that want to support TCP flows over large delay and/or large bandwidth*delay paths is the use of performance enhancing proxies, or PEPs, as illustrated in Fig. 5. Typically

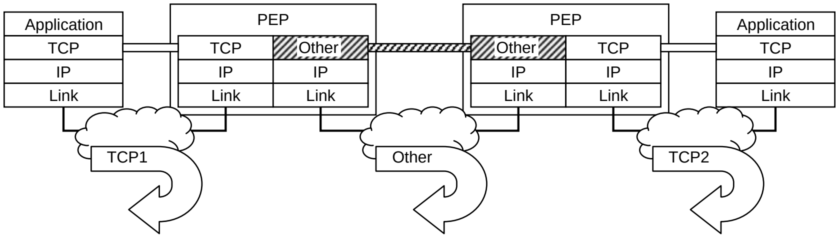


Fig. 5 Transport-layer performance enhancing proxies.

deployed as transport-layer gateways on either end of a long-delay (e.g., satellite) link, a PEP terminates an incoming TCP connection and starts up a separate transport-layer connection to the far-side PEP. That transport-layer connection can use separate parameters than the terrestrial connection, including possibly completely different congestion control mechanisms and window sizes. The far-side PEP terminates the inter-PEP transport protocol and starts up a new TCP connection to send data to the destination. This allows the end system applications to operate unmodified, since each sees a standard TCP connection with the PEP. Because the round-trip times of the connections with the end systems are low, there are generally no problems with window sizes or slow response to changes in bandwidth. If the inter-PEP connection uses dedicated bandwidth, as is often the case, then the inter-PEP transport protocol can dispense with congestion control altogether and use only rate control.

A concern with PEPs in the cislunar and other environments where there may be mobility is that PEPs are generally implemented as stateful network devices. That is, once a particular transport layer data flow begins using a pair of PEPs, it may break if the network path changes so that the pair of PEPs is no longer in the path. Thus it may not be possible to place PEPs at the most optimal locations such as at the ground station and on board the spacecraft, since a given spacecraft may need to use several ground stations to complete a given data transfer.

C. IP and End-to-End Connectivity

A second major issue with using Internet protocols for cislunar communications involves disconnection caused by either lack of communications resources or simple blockage when one endpoint is behind a moon or planet. All major IP implementations are dependent on there being an end-to-end path between the sender and receiver. If there is no active outbound connection on which to forward an IP packet at the time the routing decision is made, the packet is dropped. Also, reliable traffic in the Internet assumes bidirectional connectivity and relatively low round-trip times. Spacecraft typically have low downlink rates due to power and mass constraints, sparse and possibly simplex connectivity to Earth because of planetary alignment or resource constraints, and long round-trip times due to the large distances.

To overcome these challenging environmental conditions, a research group within the Internet Research Task Force is developing a general-purpose overlay

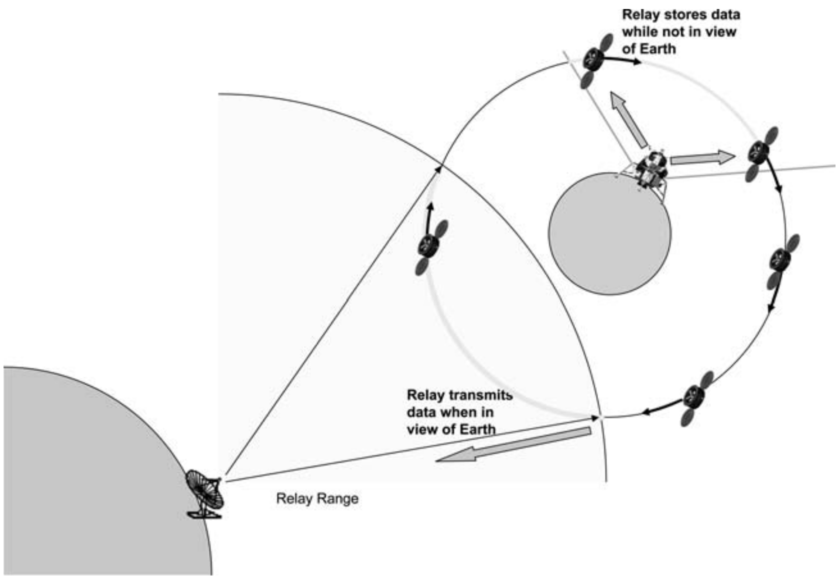


Fig. 6 Delay/Disruption Tolerant Networking example.

technology known as Delay/Disruption Tolerant Networking (DTN). In DTN, messages are transmitted among the members of the overlay network using whatever communications services are available, which could be IP, direct link-layer communication using CCSDS space packets, or other means. At each overlay node, the message is stored in persistent storage before a routing decision is made. If no active outbound connection is available, the message is held until it can be forwarded. Figure 6 shows how this can be used in a lunar environment to handle ferrying of messages by a relay orbiter from a capsule or science instrument on the far side of the moon. When on the far side of the moon, the relay orbiter (an element in the DTN overlay) picks up messages bound for Earth and, since it does not have a communications link with Earth, stores the messages. When the relay orbiter can communicate with the Earth, the stored messages are transmitted and any messages bound for the far-side object are received. While this is very similar in form to the current procedure for retrieving science data from the NASA Mars Rovers via the Mars Odyssey orbiter, DTN provides an automated mechanism for each of the data transfers. The individual messages are marked with source and destination information and are processed/forwarded automatically by the intermediate relay element(s).

VII. Conclusion

There are a number of challenges in moving from the current "individual and direct" approach to spacecraft communications to a multi-hop routed, packet multiplexed, and delay/disruption tolerant approach described here. The purported advantages in terms of simplified design, testing, and operations are all things that

either accrue over time or involve interoperability among multiple spacecraft—neither of which will be evident in the first missions. Failure to move to a more scalable communications architecture will, however, constrain our future ability to conduct the kinds of missions and campaigns discussed at the outset of this chapter.

Acknowledgments

This work draws heavily on the work of the CCSDS Cislunar Space Internetworking working group and contributions by its members. The research described here was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a Prime Contract with NASA.

References

[1] "The Vision for Space Exploration," NASA, NP-2004-01-334-HQ, 2004, http://www.nasa.gov/pdf/55583main_vision_space_exploration2-pdf.

[2] *Cross Support Reference Model Part 1: Space Link Extension Services, Recommendation for Space Data System Standards,* Consultative Committee for Space Data Systems, CCSDS 910.4-B-2, Washington, DC, Oct. 2005, http://public.ccsds.org/publications/BlueBooks.aspx.

[3] Postel, J., "Transmission Control Protocol," (IP version 4 packet format) RFC793, IETF, 1981.

[4] *Space Communications Protocol Specification (SCPS) – Transport Protocol (SCPS-TP), Recommendation for Space Data System Standards,* Consultative Committee for Space Data System CCSDS 714.0-B-1, Washington, DC, May 1999.

[5] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and Paxson, V., "Stream Control Transmission Protocol," RFC 2960, IETF, Oct. 2000.

[6] *Advanced Orbiting Systems, Networks and Data Links: Summary of Concept, Rationale, and Performance, Recommendation for Space Data System Standards,* Consultative Committee for Space Data Systems, CCSDS 700.0-G-3, Washington, DC, Nov. 1992.

[7] Postel, J., "Internet Protocol," (IP version 4 packet format) RFC791, IETF, 1981.

[8] Deering, S., and Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification," (IP version 4 packet format) RFC 2460, IETF, Dec. 1998.

[9] Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., and Weiss, H., "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Communications Magazine*, Vol. 41, No. 6, 2003, pp. 128–136.

[10] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., "An Architecture for Differentiated Service," (IP version 4 packet format) RFC2475, IETF, Dec. 1998.

[11] Nichols, K., and Carpenter, B., "Definition of Differentiated Services per Domain Behaviours and Rules for Their Specification," (IP version 4 packet format) IETF, April 2001.

[12] Braden, R., Clark, D., and Schenker, S., "Integrated Services in the Internet Architecture: an Overview," (IP version 4 packet format) RFC1633, IETF, June 1994.

[13] Rosen, E., Viswanathan, A., and Callon, R., "Multiprotocol Label Switching Architecture," (IP version 4 packet format) RFC3031, IETF, January 2001.

Chapter 6

CCSDS Space Link Extension Service Management: Real-World Use Cases

Erik Barkley* and Paul Pechkam†
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

John Pietras‡
Global Science and Technology, Inc., Greenbelt, Maryland

and

Paula Quintela§
VEGA Group PLC, Welwyn Garden City, Hertfordshire, England, United Kingdom

I. Introduction

THE Consultative Committee for Space Data Systems (CCSDS) has produced recommended standards for several Space Link Extension (SLE) data transfer services for the interoperable exchange of spacecraft telemetry and command data between spaceflight missions' ground facilities and the tracking, telemetry, and command (TT&C) networks that are used to communicate with those missions' spacecraft. As a companion activity to the specification of the SLE data transfer services, CCSDS has developed a framework for SLE service management (SLE-SM), for the creation of service management services to be used to negotiate, configure, execute, control, and monitor the provision of TT&C and SLE data transfer services [1]. In 2006, CCSDS issued the draft recommended standard (Red Book) *SLE-SM Service Specification* [2], which specifies a set of SLE-SM services through which spaceflight missions:

*Deep Space Network System Engineer.
†Deep Space Network Software Engineer.
‡Principal Engineer. Senior Member AIAA.
§Systems Engineer.

1) Submit, modify, and query the status of requests for contact periods (also known as passes).

2) Submit and query TT&C link and SLE transfer service configuration information used to fulfill requests for contact periods.

3) Submit and query Trajectory Prediction information used by the TT&C service provider to perform necessary antenna-pointing and Doppler compensation.

The SLE-SM services constitute a standard interface for the exchange of information associated with the preceding management interactions between spaceflight mission ground facilities and TT&C service providers.

The purpose of this chapter is to illustrate how the standard SLE-SM services can be applied to the operations of current TT&C service providers. It begins with a brief overview of the scope and operating environment of the SLE-SM services, and then describes several use cases in which SLE-SM services can be applied to existing operational situations. The chapter also addresses how SLE-SM services can be adopted in an evolutionary fashion. The chapter concludes with a brief identification of additions to SLE-SM that are under consideration to make SLE-SM applicable to an even broader range of network operations concepts, policies, and procedures.

II. SLE-SM Overview

A. SLE Service Management Services

The SLE-SM environment is illustrated in Fig. 1, which is adapted from the *SLE-SM Service Specification* [2]. In this model, SLE services, comprising both SLE transfer services and SLE service management, provide the interfaces between an SLE Complex that provides SLE transfer services and space link TT&C services, and a spaceflight mission that uses the services that the SLE complex provides.

The spaceflight mission is composed of a single mission spacecraft and the Mission Data Operations System (MDOS), which represents all of the mission's

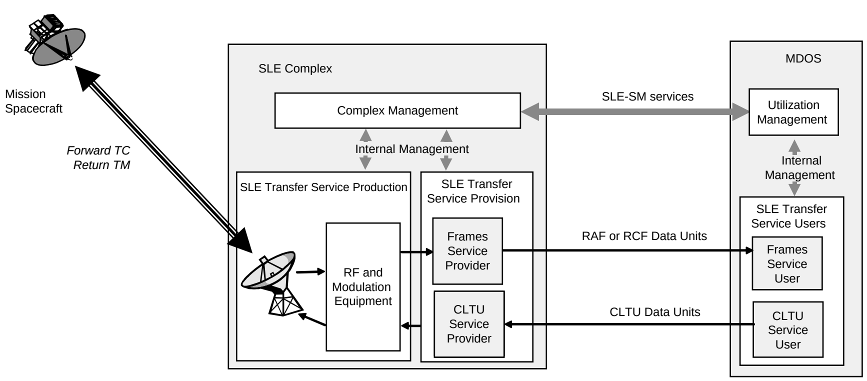


Fig. 1 SLE service management environment.

ground-based functions. A spaceflight mission uses the SLE complex's services so that the MDOS can communicate with and track the mission spacecraft.

SLE Utilization Management (UM) is the function within the MDOS that coordinates the requests by users for space link and SLE services from the complex. The UM function:

1) Coordinates radio frequency (RF), modulation, space link service, and space link extension transfer service configuration information that is used by an SLE Complex (see the following) to support subsequent requests for service.

2) Requests periods of provision of space link services and space link extension transfer services.

3) Provides Trajectory Prediction information that allows the complex to determine where the mission spacecraft will be at the requested periods of service provision.

4) Coordinates with mission user entities within the MDOS to enable the execution of SLE transfer services and to collect status information.

An SLE Complex is a collection of spacecraft TT&C-related resources under a single management authority. It may be as simple as a single ground station, or as extensive (for example) as a network of ground stations and facilities that provide the flight dynamics services associated with acquiring and maintaining communications with the mission spacecraft. SLE Complex Management (CM) is the management entity for the complex, and interfaces with UM for the following purposes:

1) Negotiating types of services, numbers of concurrent service instances allowed (where a service instance is the capability to transfer data channels of a specified type to or from a single user), and the period of performance of each Service Agreement with UM.

2) Responding to requests from the UM for individual space link sessions.

3) Providing configuration information to the resources of the complex to enable the production and provision of SLE services, and monitors their correct operation.

Because CM acts as the intermediary for UM, only those aspects of the resources of an SLE Complex that CM chooses to expose are visible to UM for management operations.

The interactions between UM and CM are the domain of SLE-SM. The other interactions illustrated in Fig. 1 are governed by other interface standards or in some cases are bilaterally determined, but in any case they are outside the purview of SLE-SM.

B. SLE Service Management Services

SLE-SM comprises a set of services for the standardized exchange of management information associated with the space link TT&C services and SLE transfer services defined in the SLE transfer service recommended standards. The management services are 1) Service Package service; 2) Configuration Profile service; 3) Trajectory Prediction service; and 4) Service Agreement service.

1. Service Package Service

The SLE-SM Service Package service is the core service of SLE-SM. The Service Package currently defines all of the space link and SLE transfer service instances

that are to be provided by a complex to a spaceflight mission for a specific duration of time. Near-term future extensions of the Service Package definition will add capabilities to specify the storage of spacecraft telemetry for subsequent *offline* playback, and to specify the offline SLE transfer service instances that will carry the played-back data. A Service Package holds information about the types of SLE transfer services to be executed [e.g., communications link transmission unit (CLTU), return all frames (RAF), and return channel frames (RCF)], the periods the services that are to be provided, the end users that access the services, the agreed configuration(s) for the space link and SLE production processes for specific space link sessions, and the Trajectory Prediction data to be referenced in provision of these services.

The Service Package service provides operations by which a UM can request that CM create, replace, delete, or change Service Packages (and operations by which CM can inform UM when Service Packages have been unilaterally modified or cancelled due to uncontrollable events).

Although the Service Package defines the space link and data transfer services to be provided during a given period of time, it does not contain all of the information within itself. Rather, it explicitly specifies the requested service provision period, but references configuration profiles and Trajectory Predictions to supply the detailed space link and transfer service configuration and orbital dynamics information (respectively) necessary to fully specify the services required.

2. Configuration Profile Service

The SLE-SM Configuration Profile service specifies a reusable set of space link and SLE services parameters that are established between UM and CM for supporting a mission spacecraft during the lifetime of the Service Agreement. Once established, a configuration profile can be referenced by any number of Service Packages. Configuration profiles allow the full set of configuration parameters to be defined independently of the Service Packages. The use of configuration profiles is well suited to the majority of spacecraft that have one or more well defined modes of TT&C operation, e.g., housekeeping, high rate instrument operation, tracking, and various combinations thereof. It allows the clear separation between the reusable configuration data of the configuration profile from the dynamic schedule information contained in Service Packages.

There are currently two categories of configuration profiles: Carrier Profiles and Event Sequence Profiles. The Carrier Profile captures RF, modulation, and coding parameters for a single carrier across the space link, as well as the configuration information for one or more SLE transfer services associated with that carrier. The Event Sequence Profile specifies a time-ordered sequence of changes to selected RF parameters to be executed by the complex. Near-term future extensions will address the offline storage and transfer service aspects of configuration profiles.

The SLE-SM Configuration Profile service is used to add, delete, and query configuration information to be referenced by Service Packages. The configuration profile service provides separate operations for handling Carrier Profiles and Event Sequence Profiles. The configuration profile service is an optional SLE-SM service.

3. *Trajectory Prediction Service*

The SLE-SM Trajectory Prediction service is used to manage the spacecraft Trajectory Prediction data that are employed by CM to derive the pointing angles and Doppler compensation settings needed to acquire the spacecraft. Using this service, UM can add, delete, and query Trajectory Prediction data that reside at CM. The Trajectory Prediction service is an optional SLE-SM service.

4. *Service Agreement Service*

The SLE-SM Service Agreement service is negotiated between the SLE Complex and the spaceflight mission to establish the service and performance envelope within which all Service Packages, configuration profiles, and Trajectory Predictions will be established during the lifetime of the relationship between the mission and the complex. The Service Agreement contains information about the services to be provided, including spacecraft communication characteristics, static and default configuration profile parameters, nominal frequency, and duration of contacts. It also specifies the range in which the exact values of parameters in Service Packages, and configuration profiles are allowed to fall, and the allowed Trajectory Prediction formats. The Service Agreement also identifies any constraints on, or specific modes of operation of, the services provided by the complex. The information contained in the Service Agreement assists the complex in determining the resources needed to support the mission (e.g., RF equipment, data storage, terrestrial network bandwidth). The Service Agreement assists UM in determining how it must construct its configuration profiles and Service Packages to make the correct and maximum use of the resources offered by the complex.

The negotiation and generation of the Service Agreement are beyond the scope of the current SLE-SM service specification recommended standard. The SLE-SM Service Agreement service has only one operation, which allows UM to query the contents of a Service Agreement. The Service Agreement service is an optional SLE-SM service.

III. SLE-SM Use Cases

A. Service Package Scenarios

The SLE-SM Service Package service allows for multiple scenarios to be defined within a single Service Package. The prime scenario represents the set of conditions [e.g., RF parameters, Trajectory Prediction information, loss of signal (LOS) and acquisition of signal (AOS) times] that will be applied unless otherwise altered by the UM. One or more alternate scenarios represent the conditions associated with preplanned contingencies, such that the SLE Complex will be "preconfigured" to support any one of those contingencies on short notice.

This particular use case involves a Service Package with two scenarios—one in reference to a predicted trajectory that has a spacecraft engine "burn" modeled into it for a trajectory correction maneuver, and the other in reference to a predicted trajectory that does not have the spacecraft engine burn modeled into it, representing the contingency situation in which the burn fails to occur as planned. For the sake of illustration of the scenario concept, this use case

presumes that the SLE Complex requires knowledge of different trajectories to maintain acquisition of the spacecraft's trajectory, e.g., for program tracking. This illustrates how preplanned contingencies can be accommodated via the SLE service management services. Figure 2 illustrates this use case. This is a simple use case demonstrating operation via the same Carrier Profile and transfer service instances across the multiple contingencies; the description will conclude with a few words about how other aspects could also vary by scenario for accommodating preplanned contingencies such as changing between RF carriers on different frequency bands.

Preconditions: For this use case it is assumed that a Carrier Profile for a return carrier (i.e., downlink) is on file with CM and has been identified as "HighGain" (in reference to the high gain antenna on the spacecraft). It is also assumed that UM has submitted two Trajectory Prediction files to CM and that they are on file with CM (see use case C for a description of how to store a Trajectory Prediction and use case D for a description of how to store a Carrier Profile at CM). The first file, given an identifier of "MRO_2006-06-16T03:15:00-Burn" has the spacecraft engine burn modeled into it; the second file, given an identifier of "MRO_2006-06-16T03:15:00-BurnFailure" does not have the engine burn modeled into it.

The use case begins with UM constructing the Service Package with two scenarios. The first scenario is given an identifier of "NormalTracking-WithBurn" (NTWB), while the second scenario is given an identifier of "ContingencyTracking-NoBurn" (CTNB). NTWB and CTNB are almost identical in data contents. The scenarios are contemporaneous and therefore indicate the same acquisition start

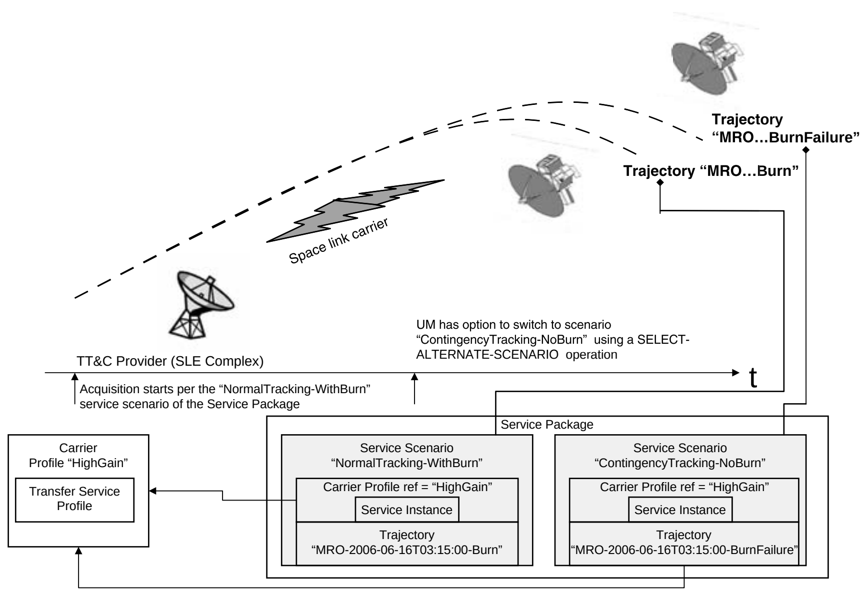


Fig. 2 Service Package with burn and no-burn alternate scenarios.

and stop times with respect to the "HighGain" Carrier Profile. Similarly, both scenarios have the same SLE transfer services specified. NTWB and CTNB differ only in the trajectories referenced; NTWB refers to "MRO_2006-06-16T03:15:00-Burn" whereas CTNB refers to "MRO_2006-06-16T03:15:00-BurnFailure". As the plan is for the tracking session represented by the Service Package to occur during the spacecraft engine burn, UM indicates this by setting the prime scenario reference of the package to be "NormalTrack-WithBurn." Having constructed the Service Package with the two scenarios, UM submits the Service Package to CM via the CREATE_SERVICE_PACKAGE (CSP) operation invocation. As a condition of accepting the Service Package, CM verifies that it can adequately provide the tracking services for both scenarios contained within the Service Package. Upon doing so, CM issues a CSP successful return to complete the operation; this indicates that CM has added the Service Package to its schedule of services.

As the Service Package enters its execution phase, CM, in conformance with the prime scenario indication, configures all of its internal equipment to provide tracking services in reference to the NTWB scenario. If, as the tracking session progresses, the Doppler signature does not match that expected with the spacecraft engine burn, but rather remains "flat," UM invokes a SELECT_ALTERNATE_SCENARIO operation for the Service Package indicating, identifying "ContingencyTracking-NoBurn" as the scenario to be executed. [For the purposes of this use case it is assumed that a method exists for UM to determine Doppler signatures, via current legacy means or (in the future) standard radiometric data and monitoring services that are being developed by CCSDS.] CM responds by reconfiguring its internal equipment to provide tracking services in conformance with the contingency scenario identified by UM and provides a return message indicating the same to UM.

Although the preceding has illustrated the use of Service Package scenarios for accommodating contingencies in relation to trajectory correction maneuvers, they can also be applied for other contingencies. For example, the scenarios may refer to different Carrier Profiles. This may be used to accommodate a contingency whereby a spacecraft enters "safe mode" operations, switching from its high gain antenna to its low gain antenna on a different frequency. If, in this example, there was some expectation that the spacecraft may be in safe mode as a result of the engine burn, a third scenario could have been added to the Service Package that referenced the "MRO_2006-06-16T03:15:00-Burn" trajectory but indicated an entirely different Carrier Profile.

B. Event Sequences

The SLE-SM Service Package service allows for preplanned event sequences to be contained within scenarios of Service Packages. Event sequences specify changes in the space link configuration as a function of time during a space link session. This type of activity occurs often in real-world operations, for example, in relation to a spacecraft's apparent elevation over a particular tracking station (i.e., better signal-to-noise ratio due to less atmospheric interference) or as a result of a spacecraft being occulted by planetary body. Event sequences have particular applicability to the deep space domain where spacecraft are often preprogrammed for space link configuration changes as a function of time and implement the

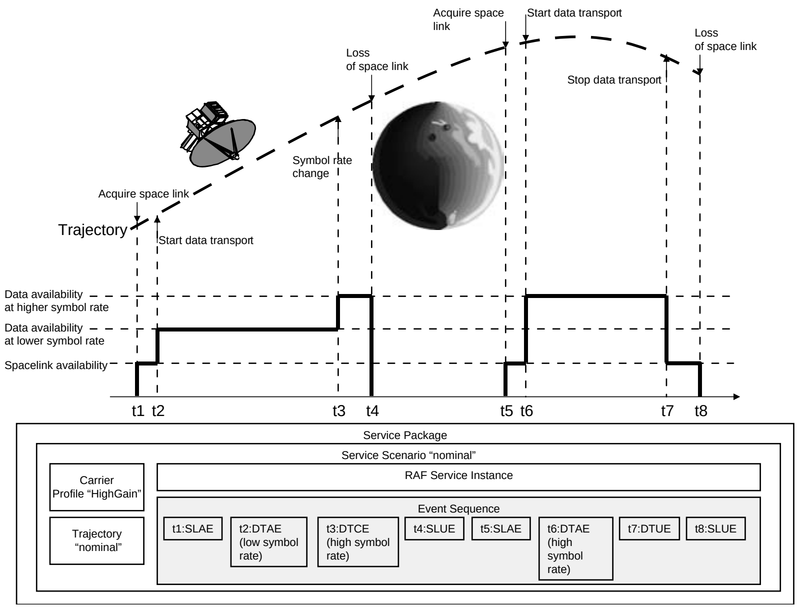


Fig. 3 Service Package with Mars fly-by event sequence.

changes in an autonomous fashion as the spacecraft are too distant from Earth for effective closed-loop, real-time control.

This use case involves a Service Package constructed to support a planetary spacecraft as it passes behind Mars on its trajectory. In this use case, days to years before the planetary encounter, flight dynamics analyses have determined when the signal will be lost and reacquired, and when the signal strengths will support various symbol rates (see Sec. IV for plans on standardizing this type of information exchange). These "events" are captured in an event sequence that is included in the Service Package for that time period. Figure 3 illustrates the Service Package that contains the event sequence for this use case.

Preconditions: For this use case it is assumed that a Carrier Profile for a return carrier (i.e., downlink) is on file with CM and has been identified as "HighGain" (in reference to the high gain antenna on the spacecraft). It is also assumed that UM has submitted a Trajectory Prediction file to CM and that it is on file with CM. (See use case C for a description of how to store a Trajectory Prediction and use case D for a description of how to store a Carrier Profile at CM.)

The use case begins with UM constructing a Service Package with a scenario containing time indexed events [in coordinated universal time (UTC), Earth receive time], based on the flight dynamics analyses. The first event, occurring at $t1$, is a space link availability event (SLAE). The SLAE indicates that the carrier from the spacecraft has been enabled. The second event, occurring at $t2$ is a data transport availability event (DTAE). The DTAE indicates that data modulation has been enabled. At $t2$, a low symbol rate applies given the spacecraft's relatively low

apparent elevation over the tracking station. At a later time, *t*3, a data transport change event (DTCE) occurs, which indicates that the symbol rate from the spacecraft will increase. The higher symbol rate is possible at *t*3 as the spacecraft's apparent elevation over the tracking station is sufficiently high to minimize atmospheric effects/interference. At *t*4, as the spacecraft is occulted by a planet, a space link unavailable event (SLUE) occurs. As the spacecraft emerges from behind the planet and is visible again over the tracking station, at *t*5, a subsequent SLAE occurs. As the spacecraft's apparent elevation over the tracking station is still sufficiently high to minimize atmospheric effects, the higher symbol rate is resumed directly at *t*6. At *t*7, the spacecraft ceases to modulate data but continues to produce the carrier tone; this is indicated by the data transport unavailable event (DTUE). At the end of the tracking session, the spacecraft is no longer visible over the tracking station and subsequent SLUE occurs at *t*8.

During the execution of the Service Package, CM utilizes the various indexed events to properly configure its internal equipment to support the acquisition of the carrier and symbol rate establishment and changes (*t*1–*t*3), confirms that the loss of signal is part of the normal tracking session (*t*4), and is prepared for the resumption of the tracking session at the proper symbol rate when those events occur (*t*5–*t*6). When the spacecraft disables the data modulation but continues to produce the carrier tone (at *t*7), CM is also prepared for this event.

It should be noted that although the Service Package service does not directly support the exchange of physical event information to assist in flight dynamic analyses for production of an indexed set of events, it does support the notion of reserving or scheduling a set of resources without stating the event sequence with the understanding that the sequence will be supplied at a later time. The Service Package service allows for this to be accomplished by invocation of a CSP operation with an "eventSequenceDeferred" setting, followed at some later time by a REPLACE_SERVICE_PACKAGE operation that contains the event sequence.

C. Trajectory Predictions

The SLE Complex uses spacecraft Trajectory Prediction data to derive the pointing angles and Doppler compensation settings needed to acquire and track the mission spacecraft. The MDOS is responsible for providing the Trajectory Prediction data necessary to support contacts.

The Trajectory Prediction service consists of the following operations: ADD_TRAJECTORY_PREDICTION (ATP), DELETE_TRAJECTORY_PREDICTION (DTP), and QUERY_TRAJECTORY_PREDICTION (QTP). SLE complexes that decide to implement the Trajectory Prediction service can select some or all of the available operations. The operations have been designed so that the MDOS is able to manage the spacecraft Trajectory Prediction data resident at the complex and hence available to support the mission contact periods. The Trajectory Prediction service handles Trajectory Prediction files in either the standard message formats defined by CCSDS, i.e., the orbit parameter message (OPM) and the orbit ephemeris message (OEM), or in a format bilaterally agreed between the space agencies involved in the cross support service. Figure 4 shows the high level organization of the Trajectory Prediction data set.

Trajectory Prediction
Trajectory prediction identifier Trajectory Prediction data (in standard or bilaterally-agreed format) Storage space remaining for Trajectory Predictions for the mission

Fig. 4 Trajectory Prediction data set.

Sometimes a spaceflight mission has its Trajectory Prediction data generated and disseminated by a separate facility from the one that invokes the Service Package service operations to create, modify, or delete the Service Packages themselves. For example, some space agencies organize their spaceflight missions so that each has its dedicated mission operations center (MOC), but multiple missions employ a common, institutional flight dynamics facility (FDF) that receives radiometric and tracking data, maintains the orbital models, and generates Trajectory Prediction for its multiple client missions. SLE-SM supports this distributed arrangement by allowing the flight dynamics facility to virtually be a part of each of the multiple MDOSs whose missions it supports. Each Service Agreement between an MDOS and a SLE Complex may permit multiple *sleSmCreatorName*s to be used by entities associated with the MDOS and/or SLE complex. The sleSmCreatorName parameter is carried in each SLE-SM message, and is used to 1) assert and control access to data sets being exchanged, and 2) identify the endpoints of various management associations within the context of a single Service Agreement.

In this use case, a standard Trajectory Prediction is to be added to the CM store of such information. As pre-conditions for this use case, it is necessary that a Service Agreement between the SLE Complex and the MDOS is in place for the spaceflight mission, and that it supports the ATP operation with standard Trajectory Prediction files. The Service Agreement also specifies two sleSmCreatorNames that can be used by the UM for mission X: one (X-MOC) that is used by the mission's MOC for invoking the various Service Package and configuration profile service operations, and another (X-FDF) that is actually given to an institutional flight dynamics facility, to which has been delegated the responsibility for invoking the Trajectory Prediction operations on behalf of mission X. Figure 5 illustrates the sequence of message exchanges among the X-FDF, X-MOC, and complex's CM.

Before the MOC requests the creation of a Service Package, it is necessary to provide to the SLE complex the Trajectory Prediction files that cover the different scenarios for the contact periods that will be included in the Service Package. To achieve that, the FDF invokes the ATP operation such that a Trajectory Prediction will be in place to support the scenarios contained in future Service Packages. Each ATP operation invocation contains a standard-formatted Trajectory Prediction, and

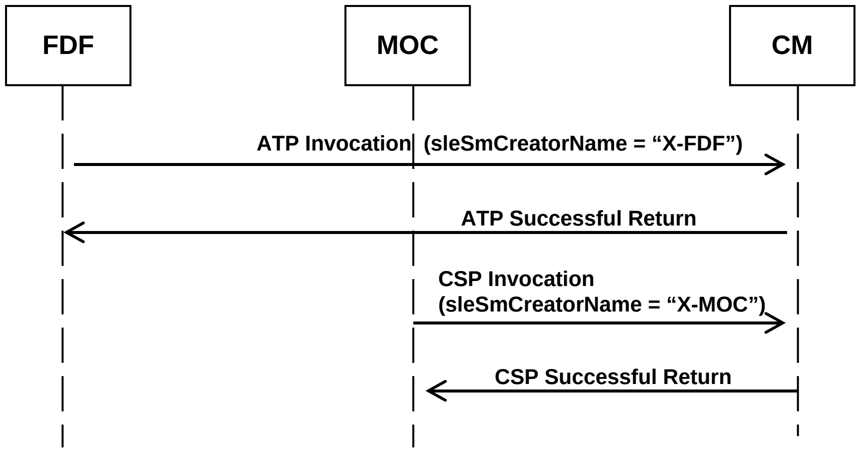


Fig. 5 Sequence of Trajectory Prediction and Service Package message exchange.

an identifier that uniquely identifies that Trajectory Prediction file with respect to the Service Agreement. The ATP operation also contains the sleSmCreatorName “X-FDF.” The sleSmCreatorName marks the resulting Trajectory Prediction file resident at the CM as being “owned” by X-FDF, and the file can be subsequently deleted only via operation invocations identifying X-FDF as the creator.

Multiple Trajectory Prediction data sets can reside at the complex for the same spacecraft for the same (or overlapping) time period(s). For example, as shown in Fig. 2, different Trajectory Prediction sets may be calculated and put in place for nominal and contingency maneuver results.

The CM validates each submitted Trajectory Prediction against constraints contained in the Service Agreement, and adds each valid Trajectory Prediction to the database of Trajectory Predictions for that spaceflight mission. At this stage, the validation performed by CM is limited; it consists mainly of checking 1) the sleSmCreatorName is permitted to invoke operations within the context of the indicated Service Agreement, 2) the storage space available at CM for the mission Trajectory Prediction files, 3) time window consistency, 4) conformance to indicated format, and 5) uniqueness of Trajectory Prediction identifier. Any required locally defined checks will be performed by CM when the Trajectory Prediction is referenced by a Service Package, during the Service Package validation. Upon successful validation, the Trajectory Prediction file is added to the CM store and an ATP successful return is provided back to the FDF.

Subsequently (minutes to months later), the MOC invokes the CSP operation to attempt to create a Service Package, using the sleSmCreatorName “X-MOC.” The CSP invocation message identifies the Trajectory Prediction that applies to each of the service scenarios in the Service Package, as illustrated in Fig. 2. As part of the validation of the Service Package, CM performs any further detailed validation of the referenced Trajectory Prediction(s). The resulting Service Package can only be replaced or deleted or have an alternate scenario selected for it via operation

invocations identifying X-MOC as the creator. Upon successful validation, the Service Package is added to the CM store and a CSP successful return is provided back to the MOC.

At service execution time, spacecraft acquisition is performed using the Trajectory Prediction data referenced by the prime service scenario within the Service Package.

D. Carrier Profiles

The use of space link Carrier Profiles to store predefined space link carrier configurations that are subsequently referenced by a Service Package is based on a well-established approach used by many of today's TTC service providers. SLE-SM supports formally defined Carrier Profiles for space links that conform to CCSDS RF and modulation standards [3], telemetry and telecommand standards [4–8], and SLE data transfer service standards [9–11]. Figure 6 shows the high-level organization of the formally defined Forward Spacelink Carrier Profile and Return Spacelink Carrier Profile.

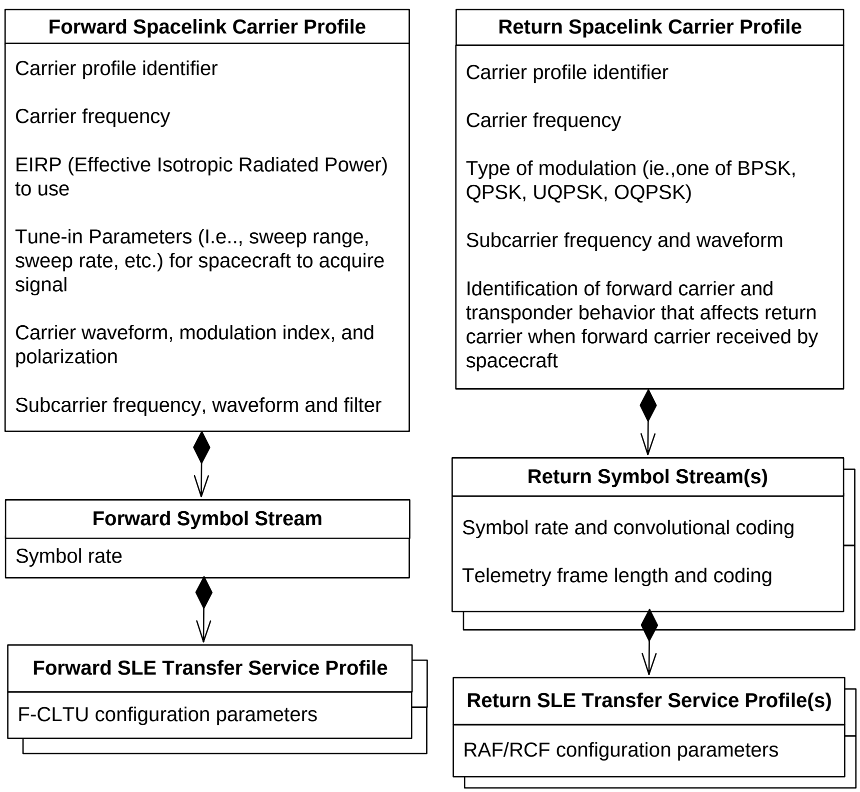


Fig. 6 High-level representation of Forward and Return Spacelink Carrier Profiles.

SLE Complexes that support the aforementioned suite of CCSDS data transfer standards may choose to implement some or all of the Carrier Profile-related operations of the SLE-SM configuration profile service: ADD_CARRIER_PROFILE (ACP), DELETE_CARRIER_PROFILE, and QUERY_CARRIER_PROFILE. When implemented, these operations provide the MDOS with fully interoperable methods for adding, deleting, and viewing (respectively) Carrier Profile information. The following use case illustrates the addition of a new Carrier Profile for a spaceflight mission that conforms to the CCSDS standards represented by the standard SLE-SM Carrier Profiles.

The preconditions for the use case include the establishment of a Service Agreement between the SLE complex and the MDOS for the spaceflight mission, and furthermore that the Service Agreement enables the ACP operation with standard Carrier Profile support. As shown in Fig. 7, at some time before the MDOS attempts to create a Service Package to be supported by the SLE Complex, the MDOS invokes the ACP operation on that complex's CM function for each forward and return carrier configuration that the MDOS expects to use in future Service Packages. Each ACP operation invocation contains a standard-formatted forward or return carrier configuration, and a Carrier Profile identifier that uniquely identifies that profile with respect to the Service Agreement. Performance of the ACP operation by the CM begins with validation of each offered Carrier Profile against constraints contained in the Service Agreement,

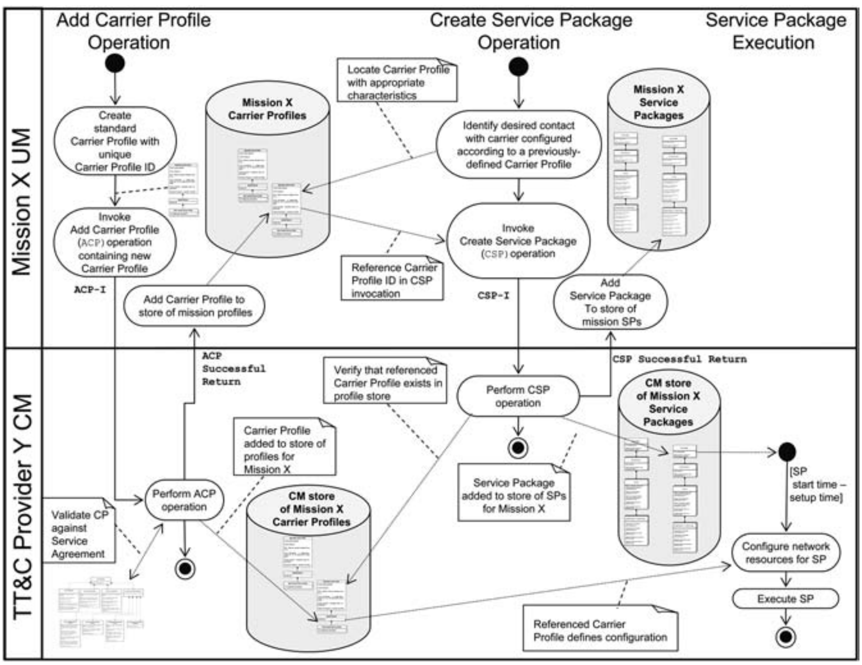


Fig. 7 Creating and executing Service Packages that reference Carrier Profiles created with the add Carrier Profile operation.

followed by adding each valid Carrier Profile to the database of Carrier Profiles for that spaceflight mission.

Subsequently (minutes to months later), the UM invokes the CSP operation to attempt to create a Service Package. The CSP invocation does not explicitly specify the desired values of the various space link and SLE transfer service configuration parameters, but rather it references one or more of the already defined Carrier Profiles using the assigned identifier(s), as illustrated in Fig. 2. If the proposed Service Package is both valid and supportable (i.e., the resources needed to fulfill the referenced Carrier Profile(s) will be available at the times requested for the Service Package), the CM schedules the Service Package and provides UM with a successful return for the operation. When the Service Package is eventually due to be executed, the complex is configured to support the contact based on the contents of the referenced Carrier Profile.

While SLE-SM provides standard operations and data structures that provide a high degree of automation and interoperability, some TT&C service providers and spaceflight missions may not be able (or may simply choose not) to use these operations. For example, a TTC service provider may offer (and a mission may use) space link services that are not fully describable by the SLE-SM Carrier Profile data sets that are part of the SLE-SM service specification, such as a non-CCSDS-standard modulation scheme. By design, the SLE-SM Service Package operations can still be used even when the Carrier Profiles do not conform to their SLE-SM standard, as described in the non-standard Carrier Profile use case.

In the non-standard Carrier Profile use case (illustrated in Fig. 8), UM and CM agree on whatever information is needed to unambiguously identify each of the carrier configurations that UM will subsequently ask CM to support in future Service Packages. In the simplest cases, these carrier configurations can be negotiated once as part of the agreement of support between the mission and the TT&C service provider. The only requirements for such carrier configurations are 1) that each one is given a unique Carrier Profile identifier, 2) each contains information that unambiguously relates the data on the space link to one or more SLE transfer service profiles, and 3) each SLE transfer service instance is given a unique profile identifier, such that it can be referenced in Service Packages. Once these carrier configurations are documented, they are incorporated into the UM and CM carrier configuration stores in the representation (e.g., database schema) used by those specific management entities. The carrier configurations are subsequently referenced (via the Carrier Profile identifiers) in Service Package operations, and used to configure service provider resources, the same way that standard-formatted Carrier Profiles are used.

While some TT&C service providers may choose not to implement the SLE-SM configuration profile service because their services cannot be fully represented by the standard Carrier Profile data sets, other providers may choose to defer implementing the configuration profile service for other reasons. For example, a service provider (or mission) may choose to gradually adopt SLE-SM, implementing at first only the (minimally required) Service Package service and relying on its existing methods for establishing and using predefined configuration information. This is expected to occur in the early adoption of SLE-SM by a number of service providers. The referencing framework of SLE-SM allows this mix of

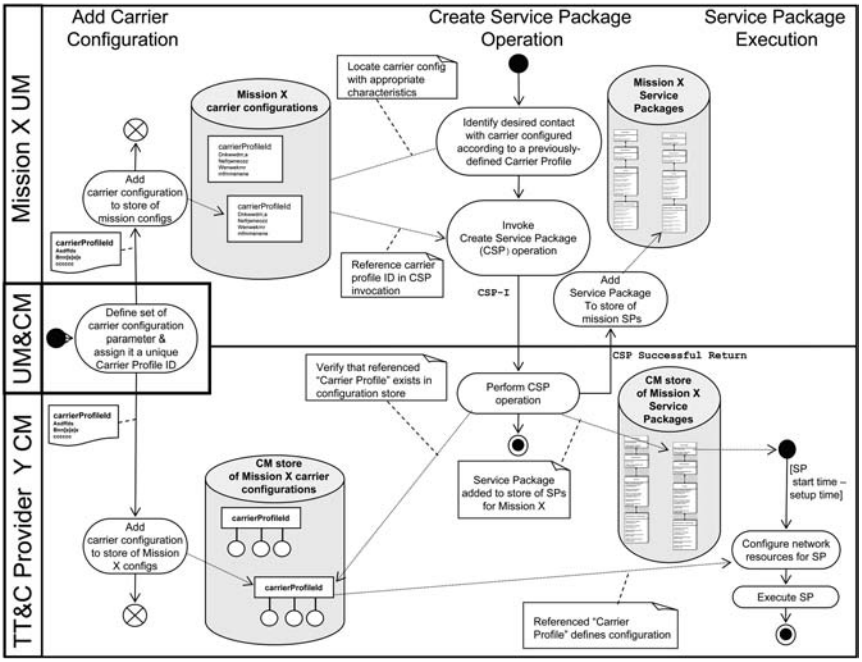


Fig. 8 Creating and executing Service Packages that reference Carrier Profiles created by local means.

standard and non-standard data sets and operations to exist as long as necessary to gracefully and incrementally adopt SLE-SM.

IV. Additional Capabilities Under Consideration

The draft recommended standard (Red-1) *SLE-SM Service Specification* defines Service Package service operations to create Service Packages by a method known as *specific scheduling*, in which the UM requests contacts with specific acquisition start time, duration, and time flexibility parameters. However, multiple space agencies today operate using what are variously known as standing orders, bulk scheduling requirements, or generic scheduling. As of the writing of this chapter, the Service Management Working Group within CCSDS uses the term *rule-based scheduling* to represent this approach to scheduling. In rule-based scheduling, long-term constraint information would be supplied in conjunction with a trajectory to a contact planning service (to be further defined), which would in turn identify opportunities for providing services consistent with the constraints and other service provider obligations. These opportunities could in turn be transformed into individually scheduled Service Packages. Examples of constraints might include requiring tracking services at least four times per week with no two contacts more than 72 h apart.

The proposed contact planning service might also include capabilities for exchanging *view period information* between UM and CM, which would identify periods of mutual visibility between mission spacecraft and appropriate complex antennas. A more sophisticated capability would couple mutual visibility with service availability, so that a UM could query a CM for available periods prior to attempting to create Service Packages in the first place.

V. Conclusion

The CCSDS has produced a draft recommendation for SLE service management services. Via the use cases presented, the ability of these services to accommodate day-to-day real-world spacecraft operations coordination with respect to TT&C service providers has been indicated. Future developments within CCSDS are aimed at providing a standard contact planning service. Completion of this activity will yield a robust and capable international standard for the necessary planning and day-to-day coordination between space missions and TT&C network service providers.

Acknowledgments

For Erik Barkley and Paul Pechkam, the research described in this chapter was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. For Paula Quintela, participation in the development of the standard described in this chapter was funded by the ESA.

References

[1] *CCSDS Recommended Standard for Space Data Systems: Cross Support Reference Model—Part 1: Space Link Extension Services*, Consultative Committee for Space Data Systems, CCSDS 910.4-B-2, Washington, DC, Oct. 2005.

[2] *CCSDS Recommended Standard for Space Data Systems: Space Link Extension—Service Management Service Specification*, Consultative Committee for Space Data Systems, CCSDS 910.1-R-1, Washington, DC, March 2006.

[3] *CCSDS Recommended Standard for Space Data Systems: Radio Frequency and Modulation Systems—Part 1: Earth Stations and Spacecraft*, Consultative Committee for Space Data Systems, CCSDS 401.0-B-16, Washington, DC, March 2006.

[4] *CCSDS Recommended Standard for Space Data Systems: TM Synchronization and Channel Coding*, Consultative Committee for Space Data Systems, CCSDS 131.0-B-1, Washington, DC, Sept. 2003.

[5] *CCSDS Recommended Standard for Space Data Systems: TM Space Data Link Protocol*, Consultative Committee for Space Data Systems, CCSDS 132.0-B-1, Washington, DC, Sept. 2003.

[6] *CCSDS Recommended Standard for Space Data Systems: TC Synchronization and Channel Coding*, Consultative Committee for Space Data Systems, CCSDS 1231.0-B-1, Washington, DC, Sept. 2003.

[7] *CCSDS Recommended Standard for Space Data Systems: TC Space Data Link Protocol*, Consultative Committee for Space Data Systems, CCSDS 232.0-B-1, Washington, DC, Sept. 2003.

[8] *CCSDS Recommended Standard for Space Data Systems: AOS Space Data Link Protocol*, Consultative Committee for Space Data Systems, CCSDS 732.0-B-1, Washington, DC, Sept. 2003.

[9] *CCSDS Recommended Standard for Space Data Systems: Space Link Extension—Forward CLTU Service Specification*, Consultative Committee for Space Data Systems, CCSDS 912.1-B-2, Washington, DC, Nov. 2004.

[10] *CCSDS Recommended Standard for Space Data Systems: Space Link Extension—Return All Frames Service Specification*, Consultative Committee for Space Data Systems, CCSDS 911.1-B-2, Washington, DC, Nov. 2004.

[11] *CCSDS Recommended Standard for Space Data Systems: Space Link Extension—Return Channel Frames Service Specification*, Consultative Committee for Space Data Systems, CCSDS 911.2-B-1, Washington, DC, Nov. 2004.

This page intentionally left blank

Chapter 7

CCSDS Standardization in Spacecraft Monitoring and Control

Mario Merri*

European Space Agency, Darmstadt, Germany

I. Introduction

THIS chapter presents a set of concepts, reference architecture, and service framework for spacecraft monitoring and control. It has been prepared by the Spacecraft Monitoring and Control (SM&C) Working Group (WG) of the Consultative Committee for Space Data Systems (CCSDS) Mission Operations and Information Management Systems (MOIMS) area.

The SM&C WG is a reasonably young initiative of the CCSDS, which was initiated in December 2003. It started as CCSDS Bird of a Feather (BOF) to investigate whether the space community had enough interest in the topic. Since then, the demonstration of interest and the participation have been significant, which justified the establishment of a dedicated CCSDS Working Group, the SM&C WG. As of today, the WG consists of the following 10 space agencies, which are actively contributing: ASI (Agenzia Spaziale Italiana), BNSC (British National Space Centre), CNES (Centre National d'Etudes Spatiales), CSA (Canadian Space Agency), DLR (Deutsches Zentrum für Luft- und Raumfahrt), ESA (European Space Agency), FSA (Federal Space Agency), INPE (Instituto Nacional de Pesquisas Espaciais), JAXA (Japan Aerospace Exploration Agency), and NASA (National Aeronautics and Space Administration).

II. Goals

A. Problem

There is a general trend toward increasing mission complexity at the same time as increasing pressure to reduce the cost of mission operations, both in terms of initial deployment and recurrent expenditure. Often, closed or "monolithic"

---

*Head of Application Mission Data System Section.

mission operations system architectures are used, which typically are not component-based in nature and do not offer open interfaces. They do not allow the redistribution of functionality between space and ground, or between nodes of the ground system. This lack of architectural openness leads to 1) lack of interoperability between agencies, 2) lack of reuse between missions, 3) increased cost of mission-specific development and deployment, 4) unavailability of commercial generic tools, 5) inability to replace implementation technology without major system redesign, and 6) lack of operational commonality between mission systems – increased training costs. The result is many parallel system infrastructures that are specific to a given family of spacecraft or operating agency, with little prospect of cross-fertilization between them.

B. Service Framework Approach

Service-oriented architecture (SOA) is an approach to system design that relies not on the specification of a monolithic integrated system, but instead on the identification of smaller, modular components that communicate only through open, published, service interfaces. As shown in Fig. 1, this framework of standard services enables many similar systems to be assembled from compliant "plug-in" components. These components may be located anywhere, provided they are connected via a common infrastructure. This allows components to be reused in different mission-specific deployments: between agencies, between missions, and between systems.

If services are specified directly in terms of a specific infrastructure implementation, then they are tied to that technology. Layering of the services themselves

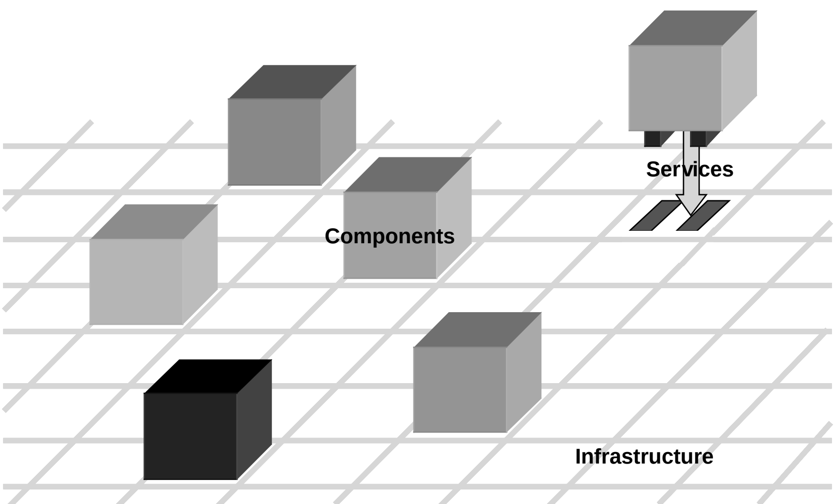


Fig. 1 Service-oriented architecture. Plug-in components communicate only via standard service interfaces through a common infrastructure. The service framework is itself layered and independent of the underlying infrastructure implementation.

allows the service specifications to be made independent of the underlying technology. Specific technology adapters enable the deployment of the service framework over that technology. This in turn makes it possible to replace the infrastructure implementation as well as component implementations. It is also possible to transparently bridge between different infrastructure implementations, where these are appropriate to different communications environments (e.g., space or ground) or simply reflect different agencies' deployment choices.

The approach is intended to be evolutionary and not revolutionary, and the service framework must also respect legacy systems. Where an integrated legacy system performs the function of several service framework components, its internal architecture and implementation does not have to be changed. Only those interfaces it exposes to other systems need be "wrapped" to make them compliant with the corresponding service interfaces. The service framework offers a range of interoperable interfaces, from which the most appropriate can be selected: compliance is not dependent on supporting them all. In this way legacy systems can be reused in conjunction with other compliant components to build a mission-specific system.

It is also important to note that the approach does not prescribe the components themselves or their implementation. Only the service interfaces between components are standardized. This allows for innovation, specialization, and differentiation in components, while ensuring they can be rapidly integrated into a system. However, for the service framework to be effective, it must ensure that semantically meaningful information associated with mission operations can be exchanged across the service interfaces, not merely data.

C. Potential Benefits

Standardization of a mission operations service framework offers a number of potential benefits for the development, deployment, and maintenance of mission operations infrastructure:

1) Increased interoperability between agencies, at the level of spacecraft, payloads, or ground segment infrastructure components.

2) Standardization of infrastructure interfaces, even within agencies, leading to reuse between missions and the ability to establish common multimission infrastructure.

3) Standardization of operational interfaces for spacecraft from different manufacturers.

4) Reduced cost of mission-specific deployment through the integration of reusable components.

5) Ability to select the best product for a given task from a range of compatible components.

6) Greater flexibility in deployment boundaries: functions can be migrated more easily between ground segment sites or even from ground to space.

7) Standardization of a limited number of services rather than a large number of specific intercomponent interfaces.

8) Increased competition in the provision of commercial tools, leading to cost reduction and vendor independence.

9) Improved long-term maintainability, through system evolution over the mission lifetime through both component and infrastructure replacement.

III. Scope

A. Mission Operations

The term mission operations is used to refer to the collection of activities required to operate spacecraft and their payloads. It includes 1) monitoring and control of the spacecraft subsystems and payloads, 2) spacecraft and ground segment performance analysis and reporting, 3) planning, scheduling, and execution of mission operations, 4) orbit and attitude determination, prediction, and maneuver preparation, 5) management of onboard software (load and dump), and 6) delivery of mission data products.

These are typically regarded as the functions of the mission control center (MCC) and are performed by the mission operations team, supported by the mission control system (MCS). Activities concerned with the exploitation of mission data, including its archiving, processing, and distribution, are considered outside the scope of mission operations. Increasingly, mission operations functions may be distributed between collaborating agencies and ground segment sites, or partially delegated to autonomous functions onboard the spacecraft itself.

The mission operations service framework is concerned with end-to-end interaction between mission operations application software, wherever it may reside within the space system. It is specifically not concerned with the provision of services for data transport or persistence (storage). It is, however, a user of such services.

B. System Boundaries and Interoperability

The needs of individual missions will require flexible collaboration between agencies. Although operational responsibility for a satellite normally resides with its owner agency, it may carry payloads, probes, or landers that are owned and operated by third parties. It is also the case that satellites from several different manufacturers may be owned and operated by a single agency. The demands for greater onboard autonomy and increasing onboard processing power will also allow migration of functionality onboard the spacecraft. This exposes more complex mission operations interactions to the space-ground interface. Standardization will enable the development of re-usable infrastructure in both ground and space segments.

Where an interface is exposed between agencies, it becomes an interoperable interface and a candidate for standardization. The variability of mission operations system configuration outlined previously means that most of the main interfunctional interfaces of mission operations could be either internal or external to a given system. Even within an agency or other operating organization, there are benefits to the standardization of mission operations services, as outlined in Sec. II.C. The concept for a mission operations service framework allows for incremental standardization as follows:

1) Priority is given to services that are currently exposed at interoperability boundaries.

2) Services exposed at key internal interfaces within the infrastructure of multiple agencies will be standardized to encourage the development of reusable infrastructure components.

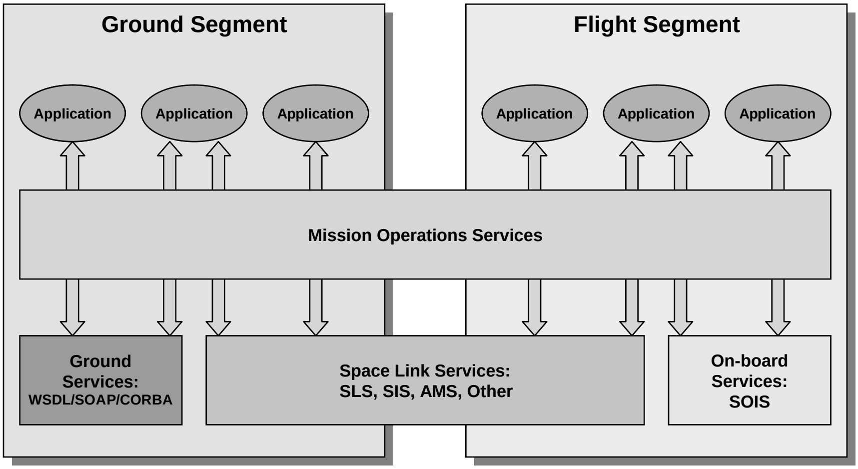


Fig. 2 Relationship of mission operations services to other CCSDS standards.

3) Finally, an initial set of services are identified to allow future evolution of interoperable interfaces with increased complexity of missions and onboard autonomy. This set is by no means predefined, and the working group anticipates that it will change as the work progresses.

C. Relationships to Other Standards

As shown in Fig. 2, the mission operations service framework addresses end-to-end interaction between applications that reside within both the space and ground segments. The underlying transport services over which mission operations services are carried may be different depending on the nature of the communications path:

1) Between space and ground. This is expected to use CCSDS Space Link Services (SLS), typically packet telemetry and telecommand (TM/TC), and optionally CCSDS Space Internetworking Services (SIS). In particular, the proposed Asynchronous Messaging Service (AMS) offers a messaging layer over which the protocol messages of the mission operations service framework could be carried. Similarly, the CCSDS File Delivery Protocol (CFDP) may be used to support file transfer.

2) Within the ground segment. Wider industry standard middleware services may be used, such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Java Remote Method Invocation (Java-RMI), Java Message Service (JMS), or Common Object Request Broker Architecture (CORBA). Alternatively, AMS could be used over Transmission Control Protocol/Internet Protocol (TCP/IP). Similarly, the file transfer protocol (FTP) may be used to support file transfer.

3) Onboard the spacecraft itself. CCSDS Spacecraft Onboard Interface Services (SOIS) could be used.

Some applications may bridge between one underlying communications environment and another, e.g., a ground mission control system may use packet

TM/TC to communicate with the spacecraft, but SOAP to forward parameter status to a payload control center. CCSDS Cross Support Services as the Space Link Extension (SLE) ones, which are not shown in the diagram, may also be used transparently to the mission operations services to extend the space link from ground stations to a mission operations center.

IV. Summary of Approach

The CCSDS SM&C Working Group has developed a concept for a mission operations (MO) service framework, which follows the principles of service-oriented architectures. It defines an extensible set of end-to-end services that support interactions between distributable mission operations functions—software applications specific to the mission operations domain.

A. Context of SMC MO Service Framework

As shown in Fig. 3, the MO service framework sits between application software specific to the domain of spacecraft mission operations and the underlying

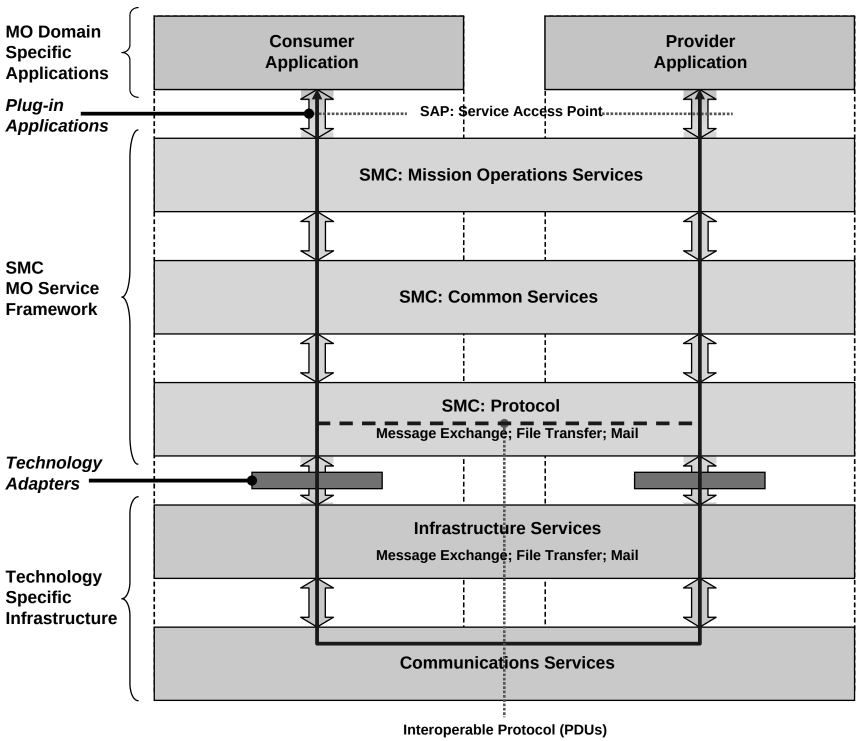


Fig. 3 Overview of SM&C mission operations service framework.

technology used for communications between distributed applications. This isolates compliant software applications both from each other and the underlying communications technology. Applications may be plugged in to the service framework through standardized service access points (SAP), one for each end-to-end service. The SAP is defined in terms of a platform independent model (PIM). For any given service, this offers two compatible interfaces that support the service consumer and service provider applications, respectively.

This approach is consistent with the Object Management Group's (OMG) model-driven architecture (MDA) approach and is capable of being fully modeled using the Unified Modeling Language (UML). Tools exist to support this approach that support and even automate the process of generating first the platform-specific model (PSM) for a given deployment technology and then the associated program code. When the SAP is bound to a specific deployment technology (e.g., programming language), it is cast as an application programmers' interface (API) specific to that technology. This API forms the interface to a reusable library that implements the MO service framework, and is called directly by the application software.

The MO service framework itself is also independent of the underlying technology-specific infrastructure. Each deployment technology requires an adapter that allows the framework to be deployed over it. This abstraction of the MO service framework from the deployment infrastructure implementation means that the entire framework can be migrated from one deployment technology to another without modification to the domain-specific applications themselves. It also allows bridging between different technologies, where these are suited to particular communications environments, or to accommodate different implementation choices between agencies.

B. SM&C MO Service Framework Layers

The SM&C MO framework has three layers, as illustrated in Fig. 4 and in the following sections.

1. SM&C Mission Operations Services

This layer provides the end-to-end services that are exposed to mission operations applications. Multiple services have been identified, each corresponding to a particular class of information that is exchanged between mission operations applications and include (non-exhaustive): core SM&C, planning, scheduling, automation, flight dynamics, time management, and onboard software management.

Each service is defined in terms of an information model that defines a set of service objects that are shared by providers and consumers of the service. Examples of such service objects are status *parameters*, control *actions*, and notification *alerts*. These constitute the basic elements of the core SM&C service. Other services concern specialized information such as orbit vectors, schedules, planning requests, and software images. In addition to definition of the static information model, the service defines the interactions required between service provider and consumer to allow 1) the service consumer to observe the status of objects through a flow of *event* messages, and 2) the service consumer to invoke *operations* upon the objects.

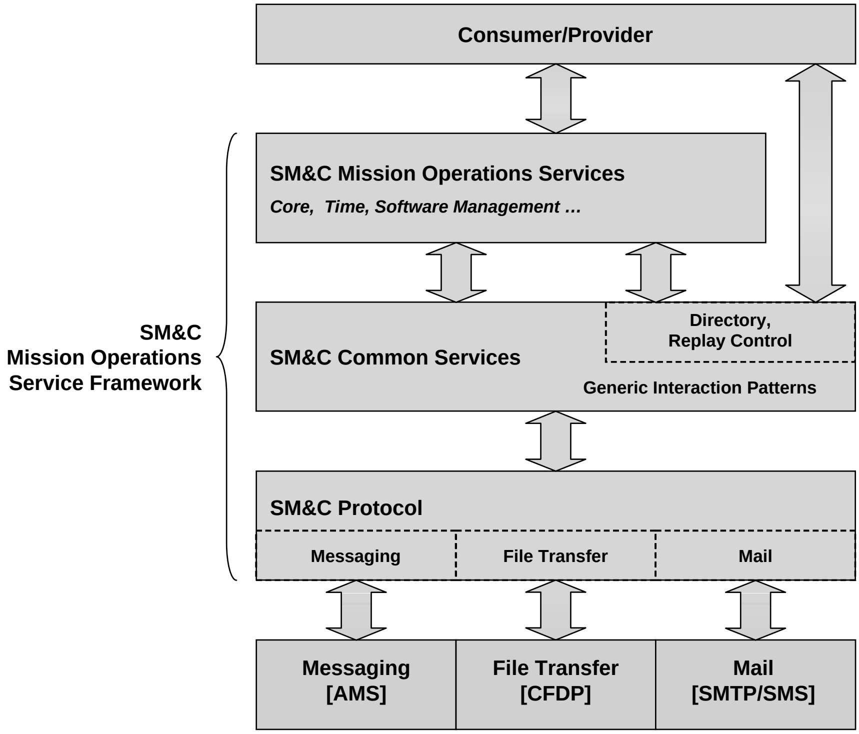


Fig. 4 SM&C service framework layers.

The service definition specifies the structure of the information objects exposed at a particular service interface. In most cases, however, each deployment (or instantiation) of a service will also require service configuration data that detail the actual service objects that exist for that service instance. For example, the core SM&C service may define what parameters, actions, and alerts are, but it is the associated service configuration data that specify the set of parameters, actions, and alerts that exist for a particular spacecraft.

2. SM&C Common Services

This layer organizes in a single place all common and generic service elements. In addition to the horizontal layering of services, the SM&C MO service framework can be broken down into a number of vertical elements. Some of these elements or subservices are common to all MO services. An example of this is the *directory service* that allows consumers to locate providers of the services they require. These common service elements are directly exposed to applications.

In analyzing the requirements for several potential end-to-end mission operations services, it became apparent that there is a lot of commonality between

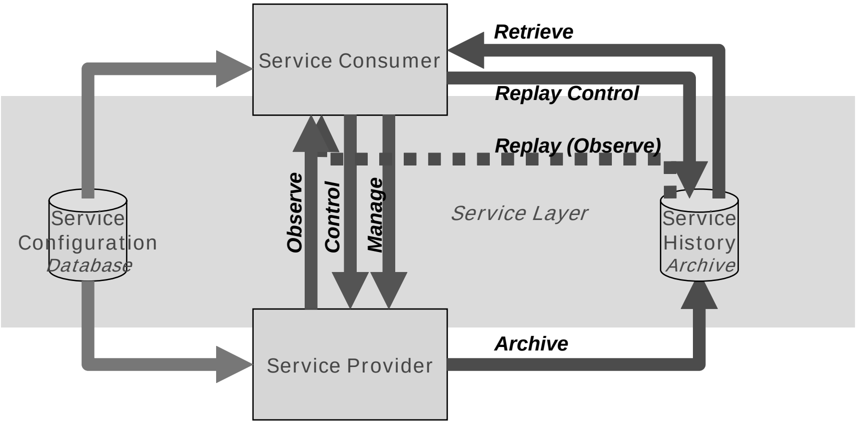


Fig. 5 Generic interaction pattern for mission operations services.

services. While the definition of objects, events, and operations are specific to the service, multiple services can be based on the same fundamental interaction pattern. As shown in Fig. 5, this includes abstract service elements such as 1) observe, interface in which the service consumer registers for interest in particular service objects and then receives status update events for those objects (e.g., obtain parameter, action, or schedule status); 2) control, interface in which the service consumer can initiate operations on objects (e.g., set parameter, send command, load schedule); and 3) manage, interface in which the service consumer can modify the behavior or processing of the service provider.

Layering of the mission operations services over these generic interaction mechanisms simplifies the task of building adapters to the underlying communications technology, as only one adapter is required to support all mission operations services. In addition to these direct interfaces, there is scope to provide generic infrastructure support for recording and retrieving service history as multiple services are based on the same generic observe interface and also for the distribution of service configuration data.

3. SM&C Protocol

The previous layers have been concerned with end-to-end interaction between service provider and consumer. Emphasis has been on the definition of service access points and the standardization of the vertical communication between the layers. Given that the implementation of the service framework itself may differ between agencies and systems, it is critical for these infrastructures to be able to interoperate such that there is standardization of the messages [or protocol data units (PDU)] that pass between provider and consumer.

The SM&C protocol layer provides this horizontal standardization between interoperable implementations of the MO service framework. The communications protocol stack beneath the MO service framework must be equivalent on both sides of the interface. Within the MO service framework itself, it is the

protocol layer that ensures interoperability, as the bindings between it and the higher layers are standardized. The protocol layer allows for three fundamental communications methods: 1) messaging (which could be implemented over AMS), 2) file transfer (which could be implemented over CFDP), and 3) mail.

It provides a thin layer within the service framework that allows separate technology adapters to be implemented for the underlying communications protocols used for each of these communications methods. Emphasis is placed on the messaging protocol, which is based on the target-controller pattern for monitoring and control. In the context of the SM&C MO service framework, the controller is equivalent to the service consumer and the target is equivalent to the service provider. A controller can be a ground control system, an onboard data handling subsystem, or a processor of a payload/subsystem. A target can be a device, a subsystem (ground based or onboard), or even an entire spacecraft. This target-controller pattern can be applied recursively.

There is a standard set of operations that the SM&C common protocol provides, that may be used to transfer directives from any controller to any target and similarly to transfer reports from any target to any controller. This standard pattern of interaction may be used to implement any of the following standard operations: 1) trigger execution of target, 2) send directive to target, 3) read state of target, 4) send indication to controller, and 5) send event to controller.

V. Conclusion

At the time of writing of this chapter, the SM&C WG has achieved 1) publication of the SM&C Green Book [1]; 2) advanced draft standards for SM&C protocol, SM&C common service, and SM&C core service; and 3) initial draft standards for SM&C time service, SM&C remote software management, and SM&C automation service. Additional information on the work-in-progress could be obtained at the Web site of the SM&C WG [2].

Additionally, the WG has developed a prototype that implements the advance standards previously discussed for validation purposes. An initial version of the prototype was successfully demonstrated in June 2006 and an upgraded version in January 2007.

References

[1] *Mission Operations Services Concept*, Consultative Committee for Space Data Systems, CCSDS 520.0-G-2, Washington, DC, Aug. 2006; also http://public.ccsds.org/publications/default.aspx.

[2] Spacecraft Monitor and Control WG Public Area, http://public.ccsds.org/sites/cwe/moims-sc.

Chapter 8

CCSDS File Delivery Protocol Performance over Weather-Dependent Ka-Band Channel

Simon S. Woo* and Jay L. Gao†

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

I. Introduction

THE success of the Mars Reconnaissance Orbiter (MRO)'s insertion on March 10, 2006, marked a key milestone in the continuation of the Mars Network which now consist of several orbiters, each performing science as well as relay services for surface assets on the red planet. In particular, the MRO will demonstrate high-capacity communications over Ka-band (32 GHz), which will greatly increase the capacity of the Mars Network. The higher capacity is critical for delivering short turnaround, operational data as well as delay-tolerant bulk science telemetry. However, data transmission in Ka-band is highly vulnerable to weather impairments. Water vapor in the atmosphere will attenuate and radiate noise that causes packet errors. In addition to the weather effects, long propagation delay in the space communication links causes further loss in signal strength and reduces the remote transmitter's ability to adapt to dynamics in the weather system.

To transfer files efficiently and reliably over long propagation space links, Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP) [1] is designed with Automatic Repeat Request (ARQ) retransmission mechanism. Unlike a terrestrial File Transfer Protocol (FTP), CFDP has four error control modes to handle the link disruptions and outages frequently encountered in space.

The usage of the CFDP retransmission mechanism is expected to take advantage of the high capacity of the Ka-band channel to provide the low latency file

---

*Member of the Technical Staff, Communication Architecture and Research Section.

†Member of the Research Staff, Communication Architecture and Research Section.

transfer with automated reliability. Although analysis of the CFDP has been studied in [2–4], the majority of these studies have not jointly considered the protocol interaction with the presence of correlated channel degradation and outages. For example, the CFDP delay was analyzed with the independent channel errors. However, weather effect was not considered in [2], and [3] has extensive discussions of the Ka-band weather phenomena, but the protocol was not incorporated. In [4], a burst error channel model was analyzed; however, it did not address the delay performance of any higher layer protocols. In [5] the bursty correlated channel was analyzed only in the terrestrial communication environment. Therefore, there is a strong need to accurately predict the CFDP file latency performance under the weather-dependent Ka-band channel, and this will be analyzed in the present study with actual Deep Space Network (DSN) data.

II. Approach and Assumptions

Although accurate characterization of the weather effects is not a trivial task, a simple modeling approach can nonetheless be taken to capture the essential aspect of the Ka-band channel and the CFDP performance by assuming that the atmospheric noise temperature is the primary source for communication errors [6]. Under this simple assumption, we ignored other factors that may contribute to the degradation of system performance such as antenna pointing loss, which in practice must be mitigated by using extra link margin. By modeling the link condition as solely driven by the atmospheric noise temperature, we isolate our analysis on the dependency between the CFDP and the weather correlation of the physical link.

In our model, we define a noise temperature threshold T_{th} that distinguishes the “good” and “bad” weather conditions. If the noise temperature is higher than the threshold, in a bad weather condition, significant error will occur. When the noise temperature is less than the threshold, in a good weather state, most of the transmitted packets will be received successfully. Naturally, in the good weather period, only small random bit error rate (BER) is assumed. On the other hand, relatively high error rate is applied to capture severe packet losses during the bad weather condition.

To analytically derive the upper bound of the delay performance of the CFDP, a two-state Markov chain is exploited to capture the weather correlations. Actual temperature data collected from the DSN Madrid site are used to find the weather statistics. The extensive use of such a model in existing research literature makes it an attractive first-cut approach for which mathematical analysis is feasible [7, 8].

There are several limitations and assumptions made throughout this work. We found that the duration of good and bad weather conditions does not fit exactly with a memoryless geometric random variable distribution, which will be shown in a later section. Therefore, actual weather data are not truly represented by the simple first-order Markov model. In general, a multistate Markov model that captures finer granularity both in terms of channel state and long-term correlation will be required for the precise modeling. Monte Carlo simulation of the protocol logic, rather than mathematical analysis, is necessary to quantify the multistate Markov chain. However, we use a two-state Markov chain for a first-cut modeling approach to get a high-level understanding of delay performance. Also, a simple two-state Markov chain allows us to derive the cumulative distribution function

(CDF) of delay and provides the theoretical foundation to analyze both the weather correlation and the protocol.

The focus of this work is to analyze the CFDP file latency at the link-layer level rather than the bit level. Thus, we will not consider analyzing the received signal strength at the physical layer. Further, to model the performance at the link-layer level, we resample the weather data on the scale of a round-trip time from Mars to DSN sites rather than the full resolution of the raw data, which is about a 1.44-minute time scale. Because the CFDP retransmission process, as illustrated later, occurs on a round-trip time scale, we condensed the noise temperature measurements to every round-trip time. That means that some dynamics in the original temperature data set will be lost. However, the following three sampling methods are explored to evaluate the impact of using the data on a coarser time scale: 1) sampling based on the average value, 2) the maximum value, and 3) the minimum value within each round-trip time interval.

III. Channel Model

In this study, the Gilbert–Elliot channel model [9] is used, where it has two states: a good and bad weather state, separated by the threshold value. In this model, most of the transmitted packets will be received successfully in the good weather state. During bad weather conditions, however, most of the transmitted packets will experience errors due to the high noise temperature at the receiving antenna. Therefore, two different BERs are applying to each good and bad weather state. We define the relatively high BER value for the bad weather state and fairly low BER value for the good weather state.

In the two-error state channel model, we assume that in a given weather state noise affects each transmitted packet identically and independently. This assumption is true when the file transmission time is short, compared to the changes in weather conditions. If we consider files that are at most 10 MB in size and a data rate about 1 Mbps for Ka-band, then the transmission time is about 80 s. As shown in [6], the sampling time is on the scale of 40 minutes. We believe it is a fair modeling approach to assume constant BER in each weather state. However, it should be noted that effects of antenna pointing error, change of the spacecraft elevation, and data rate profile during a pass will introduce variations in signal strength even under constant weather, so that a fixed BER model is only an abstraction of the practical condition. Nonetheless, we believe the independent packet error model in each state is sufficient for analysis, providing a high-level understanding of the CFDP performance over the Ka-band channel.

The probability of success and the number of transmission attempts required to complete a file transfer strongly depends on the BER and the persistence of good and bad weather states. As our analysis will show, the "burstiness" of the good and bad weather states has significant impact on the spread, or deviation, of the probability distribution of the CFDP latency.

To capture the weather correlation, the Gilbert–Elliot channel with two weather states is shown in Fig. 1. The transition from one state to another state is defined by the transition matrix P in Eq. (1), which completely characterizes the channel behavior. In this model, the current state is determined by the previous state; λ_G

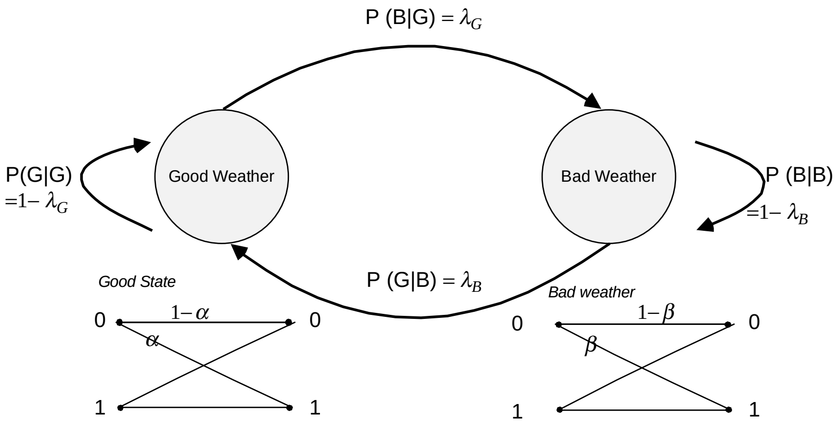


Fig. 1 Gilbert–Elliot channel model with a crossover probability α and β for the good and bad weather state.

and λ_B are the transition probabilities from a good state to a bad one and from bad to good, respectively:

$$P = \begin{bmatrix} P(G|G) & P(B|G) \\ P(G|B) & P(B|B) \end{bmatrix}$$

$$= \begin{bmatrix} 1-\lambda_G & \lambda_G \\ \lambda_B & 1-\lambda_B \end{bmatrix} \tag{1}$$

The Gilbert–Elliot channel has a characteristic that the duration of being in a good and bad weather state is geometrically distributed with mean, $1/\lambda_G$ and $1/\lambda_B$ [10]. Also, there is an independent crossover probability α and β that corresponds to the BER associated with each weather state.

IV. CCSDS File Delivery Protocol

For the CFDP file transfer, each file is divided into multiple protocol data units (PDUs) marked with a sequence number. The end-of-file PDU (EOF-PDU) is sent at the end of the initial file transmission attempt. The receiving CFDP entity, after receiving the EOF-PDU, will reply, if necessary, with a negative acknowledgment (NAK) message listing all the PDUs that were not successfully received during the initial transmission attempt. Upon the reception of this NAK message, the sender side retransmits the PDUs and the receiver will again respond with NAK messages if further retransmission is needed. This process will repeat with periodicity approximately equal to the round-trip time until all PDUs are received correctly; if all PDUs are received correctly, the receiver will send a finished message (FIN) to the sender and close the file transmission. Each round of interaction between the sender and the receiver is sometimes referred to as a spurt. The number of spurts required to complete a file transfer is a random variable, depending on the error characteristics of the channel and the file size [2]. The

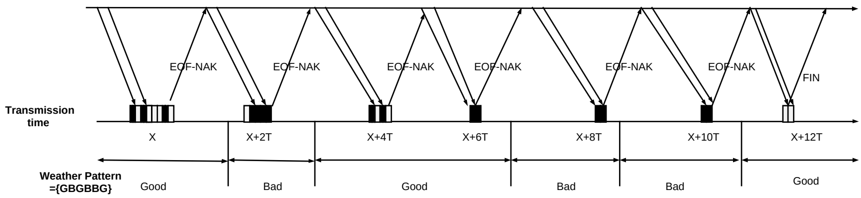


Fig. 2 Example of a CFDP file transmission.

number of spurts basically determines the number of round-trip messaging required; and, therefore, it captures the file transfer latency.

In this study, we assume the NAK message on the uplink is very reliable, and ignore the effects of protocol timers as defined in the original deferred NAK mode [1]. The timers are typically associated with the loss of the EOF and NAK messages, which we ignore to simplify our analysis. However, the loss of EOF is, on the other hand, subject to the same weather pattern as the data PDUs and therefore should be taken into account in follow-on studies. To better understand the CFDP operations, an example is given in Fig. 2. In Fig. 2, with the given weather pattern {GBGBBG}, T is defined as a one-way trip time from the sender to the receiver, which could be as long as 20 minutes X represents the time (minutes) of initial PDU receptions. In this example, the file is composed of 8 PDUs and 7 spurts are required to receive all PDUs correctly.

V. Weather Statistic and Sampling

The atmospheric noise temperatures are collected at the DSN Madrid for 5356 days. There are some inconsistencies and gaps in the data set; hence, preprocessing is required to acquire a subset of data with a consistent time interval between each data point. After preprocessing, 1956 days of data consisting of 1,532,456 temperature measurements are obtained, where every data point is almost 1.44 minutes apart.

Because of the long propagation delay, the time difference between the previous file transmission attempt and the current file transmission attempt is relatively large compared to the original noise temperature data collection time interval. For the link layer perspective, it is more meaningful to get the weather statistics based on every round-trip time interval than 1.44 minutes interval. The weather condition at the instance of PDUs' arrival at the DSN antenna is the most important time for PDUs' successful receptions. Therefore, there is a good reason to resample data at the resolution of round-trip time rather than simply using the 1.44 minute interval. Specifically, we assume a 40-minute round-trip time from Mars to the DSN, so that the factor by which the data set is condensed is given by

$$\text{Number of data points} = \frac{\text{Round-trip time}}{\text{Sample interval}} = \frac{40\text{ min}}{1.44\text{ min}} = 27.78 \tag{2}$$

As shown in Eq. (2), we need to sample once every 28 data points in the original data set. However, some of the weather information will be lost. To capture the envelope on the impact of under-sampling the data, we processed the data in three different ways: 1) average every 28 samples and use the average value, 2)

select the maximum reading among the 28 samples, and 3) choose the minimum reading of the 28 samples. The reasoning behind averaging 28 samples is to compress and compact the original data points, while still capturing the relative contribution of each data point; choosing the maximum and minimum sample should provide the upper and the lower bound to evaluate the potential deviation due to under-sampling of the data set.

Once temperature data are sampled to round-trip time scale, a noise temperature threshold T_{th} can be defined in the following way to convert the temperature data into two groups:

$$\begin{cases} T > T_{th}, \text{ bad weather, } (W = B) \\ T \leq T_{th}, \text{ good weather, } (W = G) \end{cases} \tag{3}$$

If the sampled noise temperature is greater than the threshold, we define the weather as being in the bad state, and denote it as $W = B$. Otherwise, we define weather as in the good state and represent it as $W = G$. The statistics of the sampled data by averaging, maxima, and minima are shown in Table 1 with the threshold of 20 Kelvin (K).

To describe the probability of encountering a good weather condition, *availability* metric is defined as the percentage of time it is capable of providing services [11]. Thus, availability can be calculated as a percentage of good weather in the preprocessed DSN temperature data, and the 20 K threshold roughly corresponds to the 89% weather availability shown in Table 1. We fixed the threshold value, since usual data transmission condition is around 90% availability. With a 20 K threshold, the statistics of the temperature data obtained from three different sampling methods are recorded in Table 1.

From Table 1, we found that the sampling with averaging produces a fair match with the availability and mean obtained from the original weather data set. Sampling the maximum and minimum data points cause significant deviation from the statistics of the original data set shown in Table 1.

VI. Estimating Parameters from Sample Statistics

Once weather data are sampled in every round-trip time, the average durations of the consecutive good and bad weather are tabulated. Our goal is to approximate the

Table 1 Statistics of the temperature data sampled in every round-trip time interval with the threshold = 20 K

	Preprocessed original data	Sampling data by averaging	Sampling data by selecting maximum	Sampling data by selecting minimum
Availability	89.42%	88.04%	82.47%	92.54%
Data size (number of samples)	1532456	54730	54730	54730
Maximum, K	270.5059	214.76	270.5059	159.809
Minimum, K	5.3324	5.9289	6.4332	5.3224
Mean, K	16.1889	16.19	19.6919	13.9116
Std	14.8673	14.1322	22.2187	9.5796

lengths of the consecutive good and bad weather as a memoryless geometric random variable so that a two-state Markov chain can be applied. With data obtained from three different sampling methods, each CDF of the number of consecutive good and bad weathers is plotted. In Fig. 3, the theoretical CDF of the geometric random variable with the mean ($1/\lambda$) obtained in Table 1 is overlaid with the CDF obtained from

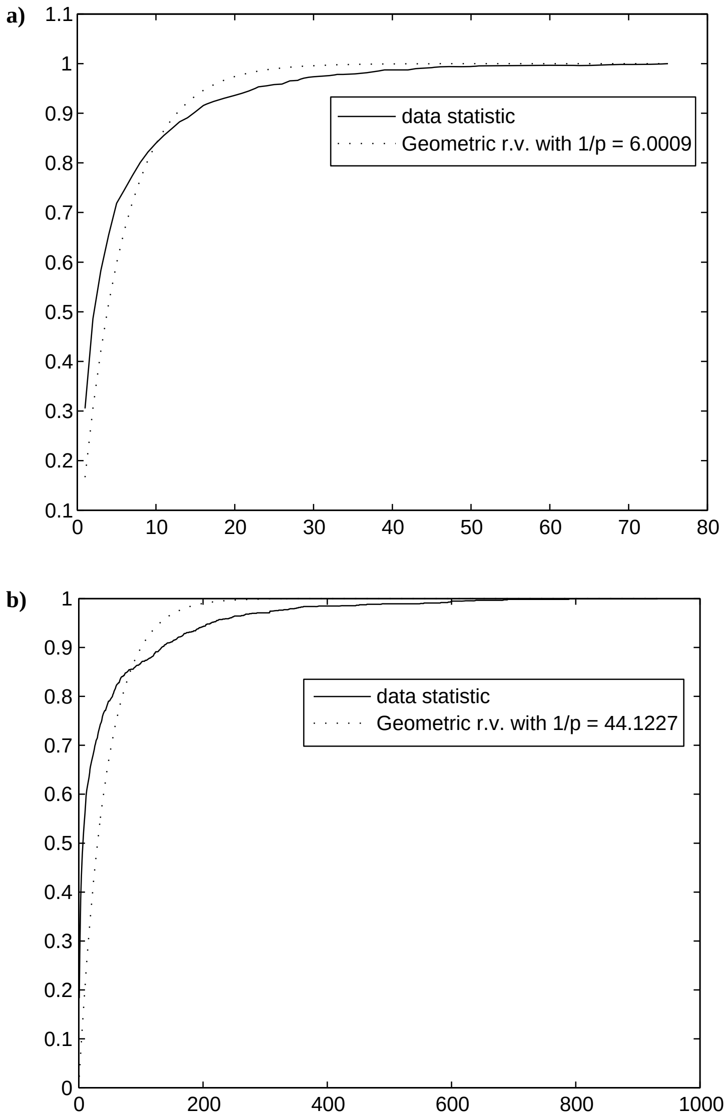


Fig. 3 CDF of bad and good weather with 20 K threshold. a) Bad weather; b) good weather.

Table 2 Duration of bad and good weather statistic by averaging, choosing maximum and minimum sampling with threshold = 20 K

	Bad weather			Good weather		
	Sampling by averaging	Sampling by choosing the maximum	Sampling by choosing the minimum	Sampling by averaging	Sampling by choosing the maximum	Sampling by choosing the minimum
Maximum duration	75	111	74	932	600	1771
Minimum duration	1	1	1	1	1	1
Mean duration	6.0009	6.1817	5.5586	44.1227	29.06	68.1902
Std	8.7483	9.9479	8.0311	95.5634	66.1582	162.1346

the sampled weather statistic. It is noted that these two CDFs do not align very well; there are noticeable deviations, especially for the good weather distribution. Therefore, to accurately model the performance of the CFDP, more states will typically be required. However, the geometric random variable assumption provides the simple way to analyze the complicate weather events. Therefore, in the present scope of the study, we model the good and bad weather processes with a geometric distribution.

The durations of the good and bad weather statistic by averaging, maxima, and minima with a threshold of 20 K are recorded in Table 2. As expected, sampling by selecting the maximum yields the longest average duration of the bad weather and the shortest average good weather duration. Opposites are true with the minimum data sampling method, as expected. The statistics obtained from sampling by averaging reside between the two extreme sampling methods. The data obtained from Table 2 is consistent with the result from Table 1. From Table 1, we can observe that there are more statistical variations in the good weather distributions. This is partly because most of the temperature data are below 20 K, which are translated into good weather data points. Hence, how we sample the original data can alter the original good weather data statistic. On the other hand, there are quite fewer bad weather data points; therefore, the bad weather statistic is less sensitive to the choice of data sampling method. The relationship between the selection of data sampling methods and the delay performance is evaluated in a later section. The duration unit in Table 2 is 40 minutes, because we sampled data every 40 minutes.

As defined in an earlier section, the transition probability from good to bad states can be derived by 1/(average duration of good weather) and the transition probability from bad to good state is 1/(average duration of bad weather). The transition probabilities of different sampling methods with the threshold of 20 K are shown in Table 3. These transition probabilities will be used for simulating weather patterns.

Table 3 Transition probabilities with threshold = 20 K

	Sampling by averaging	Sampling by choosing the maximum	Sampling by choosing the minimum
$P(G\|G)$	0.9773	0.9656	0.9853
$P(B\|G)$	0.0227	0.0344	0.0147
$P(G\|B)$	0.1667	0.1618	0.1799
$P(B\|B)$	0.8333	0.8382	0.8201

VII. Mathematical Analysis

In this section, the expression of the CDF of the number spurts required to complete the file transmissions is derived and evaluated. The CFDP file delivery latency D_f is given by

$$D_f = T \cdot [1 + 2(N_s - 1)] \tag{4}$$

where T is the one-way propagation delay and N_s is the random variable representing the number of spurts required to transfer a file completely and correctly.

Let N be the number of PDUs in the file, $N_{tx}(i)$ be the number of transmissions required to send the ith PDU, and w_k be the kth weather pattern, which is a single realization of weather pattern to measure the CFDP file transfer until success. Then, the CDF of N_s is defined as follows:

$$P[N_s \leq m] = \sum_{k=1} P[N_s \leq m \mid w_k] \cdot P[w_k] \tag{5}$$

where $w_1, w_2, \ldots w_k, \ldots$ are mutually exclusive weather patterns. By applying the theorem of total probability [10], Eq. (5) can be obtained.

However, evaluating the preceding expression over all possible realizations of weather patterns cannot be done manually for the potentially large weather patterns. Therefore, we rely on numerical methods to generate the realizations of weather patterns, where each weather pattern is produced according to the transition probabilities in Table 2.

Once a weather pattern is generated, the analysis follows similarly with the independent channel in [2]. For a given weather pattern, the noise affects each PDU independently. Therefore, the number of transmissions required to deliver the ith PDU, $N_{tx}(i)$, is independent with the transmission of the jth PDU, $N_{tx}(j)$.

It is observed that the number of spurts required to complete a file transfer, N_s, is simply the maximum number of transmissions required among all PDUs in the file. Therefore, N_s can be expressed in the following way:

$$N_s = \max_{i \in \{1,\ldots,N\}} \{N_{tx}(i)\} \tag{6}$$

By substituting Eq. (6) into Eq. (5), the following equation can be obtained:

$$P[N_s \leq m \mid w_k] = P\left[\max_{i \in \{1,\ldots,N\}} \{N_{tx}(i)\} \leq m \mid w_k\right] \tag{7}$$

The noise affects each PDU independently in the given weather patterns, and Eq. (7) becomes

$$P\left[\max_{i\in\{1,\ldots,N\}}\{N_{tx}(i)\} \le m \mid w_k\right] = P[N_{tx}(1) \le m \mid w_k]\cdot P[N_{tx}(2) \le m \mid w_k]\cdots\cdot P[N_{tx}(i) \le m \mid w_k]\cdots\cdot P[N_{tx}(N) \le m \mid w_k] \tag{8}$$

Since $N_{tx}(i)$ is independently identically distributed (i.i.d.), Eq. (8) can be written into the product form as follows:

$$\prod_{i=1}^{N} P[N_{tx}(i) \le m \mid w_k] = \{P[N_{tx}(1) \le m \mid w_k]\}^N = \left\{\sum_{j=1}^{m} P[N_{tx}(1) = j \mid w_k]\right\}^N \tag{9}$$

If we expand each term in Eq. (9), we obtain

$$\{P[N_{tx}(1) = 1 \mid w_k] + \cdots + P[N_{tx}(1) = i \mid w_k] + P[N_{tx}(1) = i+1 \mid w_k] + \cdots + P[N_{tx}(1) = m \mid w_k]\}^N \tag{10}$$

Let us introduce

$$P(i, k) = P[N_{tx}(1) = i \mid w_k] \tag{11}$$

as the probability of the first PDU transmission being successful in the ith transmission attempt given the kth weather pattern. Using Eq. (11), Eq. (9) can be rewritten as

$$\left\{\sum_{j=1}^{m} P[N_{tx}(1) = j \mid w_k]\right\}^N = \{P(1, k) + P(2, k)\cdots + P(i, k) + P(i+1, k) + \cdots + P(m, k)\}^N \tag{12}$$

In Eq. (12) $P(i+1, k)$ can be computed from the following recursive relation:

$$P(i, k) = \begin{cases} P'(i,k)\cdot(1-p), & w_k(i) = G \\ P'(i,k)\cdot(1-q), & w_k(i) = B \end{cases}$$

$$P'(i,k) = \begin{cases} P(i,k)\cdot\dfrac{p}{1-p}, & w_k(i) = G \\ P(i,k)\cdot\dfrac{q}{1-q}, & w_k(i) = B \end{cases} \tag{13}$$

$$P(i+1, k) = \begin{cases} P'(i,k)\cdot(1-p), & w_k(i+1) = G \\ P'(i,k)\cdot(1-q), & w_k(i+1) = B \end{cases}$$

where i and k are the positive integer and $w_k(i)$ is the ith weather condition in the kth weather pattern, and p and q are the PDU error rates for good and bad weather. $P'(i, k)$ is the probability of transmission failure up to ith transmission attempts in the kth weather sequence. $P'(i, k)$ is, therefore, the intermediate term to calculate

the next $P(i+1, k)$ term through the recursive relationship, assuming the ith PDU transmission has failed.

As an example, the first two terms in Eq. (12) are calculated:

$$P(1, k) = \begin{cases} (1-p), & w_k(1) = G \\ (1-q), & w_k(1) = B \end{cases}$$

$$P'(1, k) = \begin{cases} P(1,k) / \dfrac{p}{1-p}, & w_k(1) = G \\ P(1,k) / \dfrac{q}{1-q}, & w_k(1) = B \end{cases}$$

$$P(2, k) = \begin{cases} P'(1,k) \cdot (1-p), & w_k(2) = G \\ P'(1,k) \cdot (1-q), & w_k(2) = B \end{cases} \tag{14}$$

Let us introduce $f(i)$ and $g(i)$ to further simplify the terms in Eq. (14):

$$f(i) = \begin{cases} \dfrac{p}{1-p}, & w_k(i) = G \\ \dfrac{q}{1-q}, & w_k(i) = B \end{cases}, \quad g(i) = \begin{cases} 1-p, & w_k(i) = G \\ 1-q, & w_k(i) = B \end{cases} \tag{15}$$

Note that $f(i)$ and $g(i)$ will greatly simplify Eq. (12) with single weather state case.

Using Eq. (15), Eq. (12) can be written in the functional form as follows:

$$\begin{aligned} P[N_s \leq m \mid w_k] &= \{P(1, k) + P(2, k) + \cdots + P(i, k) + P(i+1, k) + \cdots + P(m, k)\}^N \\ &= \{P(1, k) + P'(1, k) \cdot g(2) + \cdots + P'(i-1, k) \cdot g(i) + P'(i, k) \cdot g(i+1) \\ &+ \cdots + P'(m-1, k) \cdot g(m)\}^N = \{P(1,1) + \cdots + P(1,1) \cdot f(1) \cdot g(2) \cdot \cdots \cdot f(i-1) \\ &\cdot g(i) + \cdots + P(1,1) \cdot g(2) \cdot \cdots \cdot f(m-1) \cdot g(m)\}^N \end{aligned} \tag{16}$$

Equation (16) is the expression of the CDF of the number of transmissions required to complete a file transfer over the kth weather sequences. To quantify the CDF, we apply a numerical computation method that iterates over a very large set of weather patterns.

The number of iterations required to obtain the reliable CDF can be found in Hoeffding's Inequality [12]. In this work, lengths of 1,000,000 random good and bad weather realizations are generated according to the transition probabilities found in Table 3. The CFDP simulation starts at a random location once the weather sequences are generated long enough to be in the stationary distribution. Four thousands different weather sequences are generated with the length of each sequence to be 1,000,000.

VIII. Evaluation of the CFDP Latency

In this section, the CDF is evaluated over various BERs, and file sizes over three different data sets obtained from average, maxima, and minima sampling methods. Computation utilizing the parameters in Table 4 is conducted to evaluate the CDF of the CFDP file latency. A single PDU size is fixed as 1 KB with

Table 4 The CDF evaluation parameters

Simulation parameters	Values
Round-trip time from Mars to Earth (minutes)	40 minutes
PDU size	1 KB
BER in good weather	10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}
BER in bad weather	10^{-3}, 10^{-4}
File size	1 MB, 10 MB
Availability	88%
Data sampling method	Average data sampling, maximum data sampling, minimum data sampling

file size at either 1 or 10 MB. BERs of 10^{-5}, 10^{-6}, 10^{-7}, and 10^{-8} are considered for a good weather state; 10^{-3} and 10^{-4} BERs are used for the bad weather state; 10^{-3} and 10^{-4} BER corresponds to the 99 and 55% of PDU error rate, respectively, as shown in Table 5.

The overall goal of this evaluation is to find the CDF of the maximum number of spurts required to successfully complete the file transfer. The CDF plot captures the dynamics in the behavior of the CFDP over the weather-dependent link. The CDF provides design guidelines for mission planners who wish to have quantitative understanding on percentile upper bound on the file transfer latency when using the CFDP. Besides the upper bound of the delay performance, the average number of transmissions required is another major interest because it represents the long-term average performance of the system.

A. Effect of Applying Different BERs in Good and Bad Weather

The effects of using different BERs on the fixed file size with data obtained from the average sampling methods are examined. We observed that reducing BER in bad weather significantly improves the 99% file transmissions requirement and this quantitatively indicates that taking advantage of good weather conditions is extremely important for Ka-band file transmissions.

Table 5 BER and the corresponding PDU error rate

BER	PDU error rate
10^{-3} (bad)	0.9997
10^{-4} (bad)	0.5507
10^{-5} (good)	0.0769
10^{-6} (good)	0.008
10^{-7} (good)	7.99968×10^{-4}
10^{-8} (good)	7.9997×10^{-5}

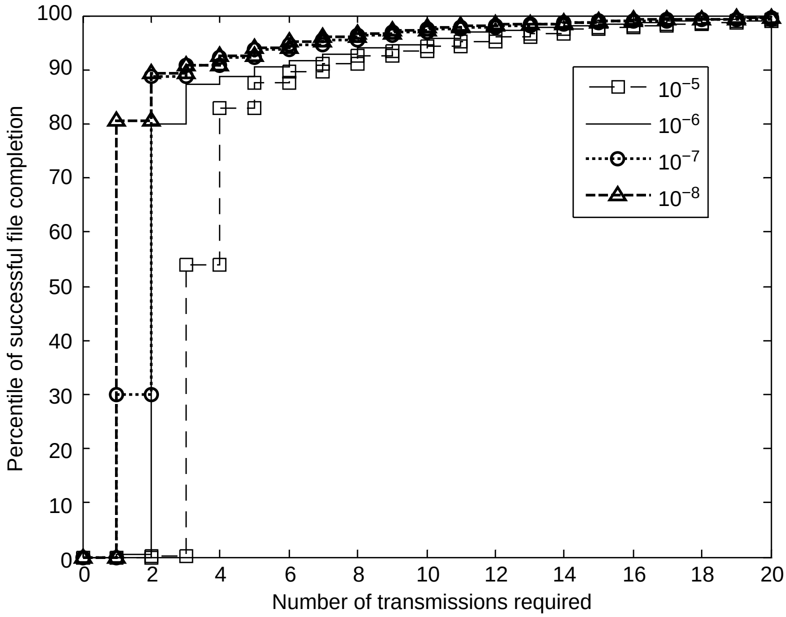


Fig. 4 10 MB file latency performance with 10^{-3} bad weather BER.

To easily refer to the data points in the figures, we introduce a parenthesis notation to indicate the good and bad weather BER pair. For example, (a, b) indicates that a is the BER at good weather, and b is the BER at bad weather.

To achieve the 99% file completion, the maximum numbers of transmission required are 20, 17, 16, and 15 for (10^{-5}, 10^{-3}), (10^{-6}, 10^{-3}), (10^{-7}, 10^{-3}), and (10^{-8}, 10^{-3}) BER pairs, as shown in Fig. 4. Notice that reducing the BER in good weather conditions enhances the required number of spurts by 25%, from 20 to 15 transmissions.

The results for 10^{-4} bad weather BER are recorded in Fig. 5. The maximum number of transmissions required to complete the 99% of file transfers are all 12 transmissions for (10^{-5}, 10^{-4}), (10^{-6}, 10^{-4}), (10^{-7}, 10^{-4}), and (10^{-8}, 10^{-4}) BER pairs. By reducing the bad weather BER from 10^{-3} to 10^{-4}, the maximum number of transmissions required to achieve the 99% file completion goes down from 20 to 12 transmissions with the 10^{-5} good weather BER; the maximum number of transmissions goes down from 15 to 12 transmissions with 10^{-8} good weather BER. The gain is achieved from the fact that about half of the PDUs are received correctly in the 10^{-4} bad weather BER case, whereas almost all PDUs are lost in the 10^{-3} bad weather BER.

Although the maximum number of transmissions required to achieve 99% file completion is more than 10, the average number of transmissions are much fewer. Using the following equation, the average number of transmissions required to complete the 99% of file transfer is computed:

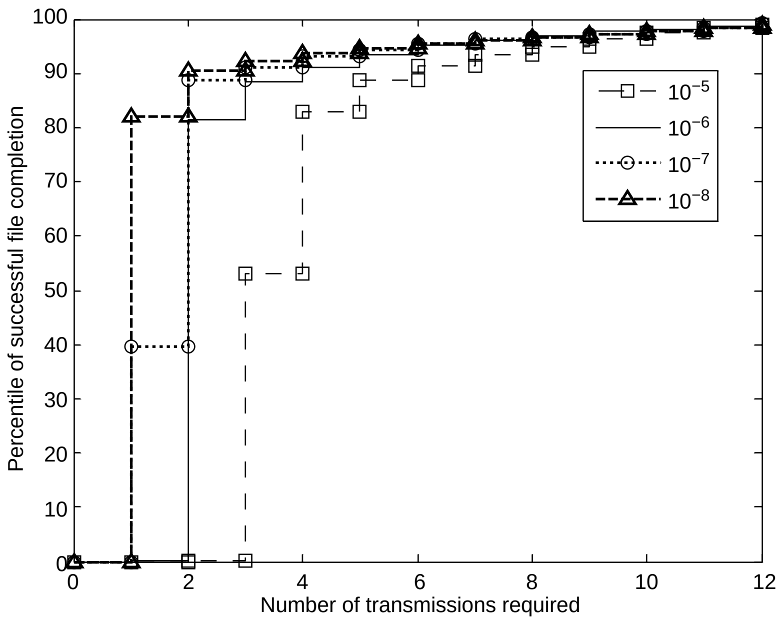


Fig. 5 10 MB file latency performance with 10^{-4} bad weather BER.

$$\text{E [number of transmissions]} = \sum_{k=1} P[k] \cdot k \tag{17}$$

where k is the number of transmissions required.

For 99% of the file completion, the average number of transmissions are 4.19, 2.91, 2.23, and 1.73 for (10^{-5}, 10^{-3}), (10^{-6}, 10^{-3}), (10^{-7}, 10^{-3}), and (10^{-8}, 10^{-3}) BER pairs, as shown in Fig. 6. By reducing the BER from 10^{-5} to 10^{-8} in good weather, at most 2.46 transmissions can be reduced on average, as shown in Fig. 6. For each (10^{-5}, 10^{-4}), (10^{-6}, 10^{-4}), (10^{-7}, 10^{-4}), and (10^{-8}, 10^{-4}) pair, the average number of transmissions are 3.92, 2.54, 2.00, and 1.54. A moderate performance gain can be achieved on average by reducing a BER for both good and bad weather from (10^{-5}, 10^{-3}) to (10^{-8}, 10^{-4}), which saves up to 2 transmissions.

B. Effect of Different File Sizes

In this section, a 1 MB file size is considered with the same parameters. The CDFs of the number of transmissions required for 99% file completion are plotted in Figs. 7 and 8. From Figs. 7 and 8, we can see that the smaller file size improves latency performance. The order of improvement can be found by comparing Figs. 4 and 5 with Figs. 7 and 8. The 1 MB file size saves up to two transmissions, or equivalently about two round-trip times, for 99% file completion.

The average number of transmissions required is shown in Fig. 9. The differences of the average numbers of transmissions required with the 10 MB and 1 MB

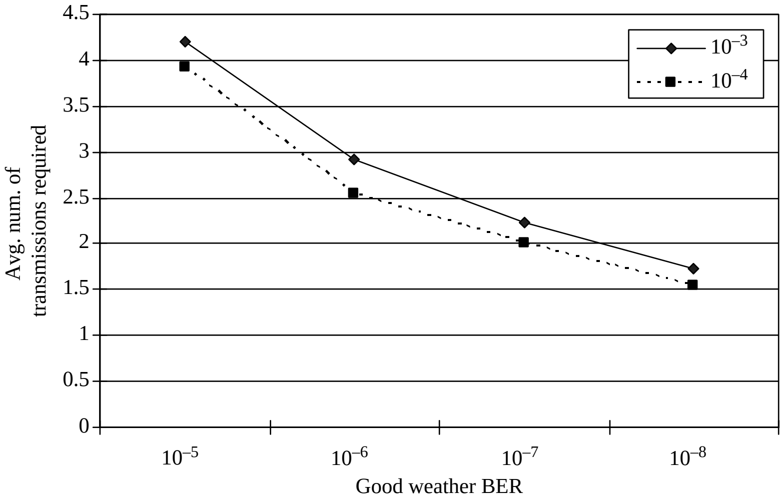


Fig. 6 Average number of transmissions required to achieve the 99% file completion of 10 MB file size.

file sizes are plotted in Fig. 10 in the 10^{-3} and 10^{-4} bad weather BER. It is shown that, on average, about 3.2 transmissions and 1.5 transmissions are required to transmit a 1 MB file for 10^{-3} and 10^{-4} bad weather BER, respectively. From Fig. 10, we can see that reducing the file size has the performance gain ranging

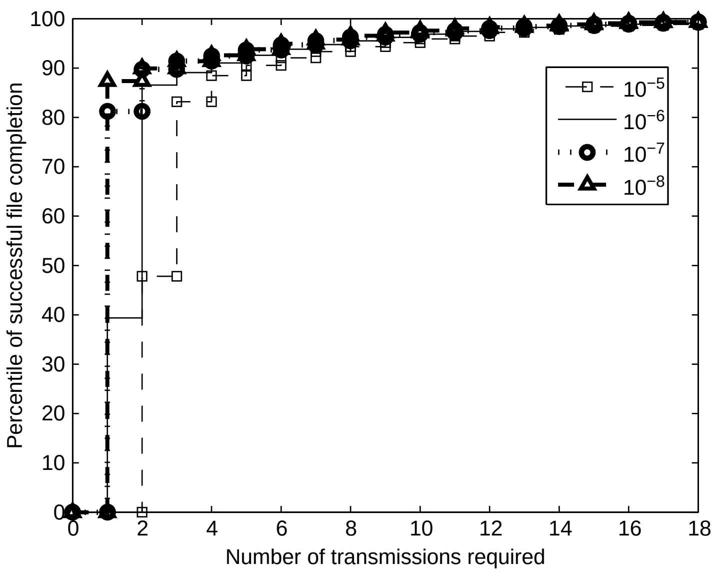


Fig. 7 1 MB file latency performance with bad weather BER at 10^{-3}.

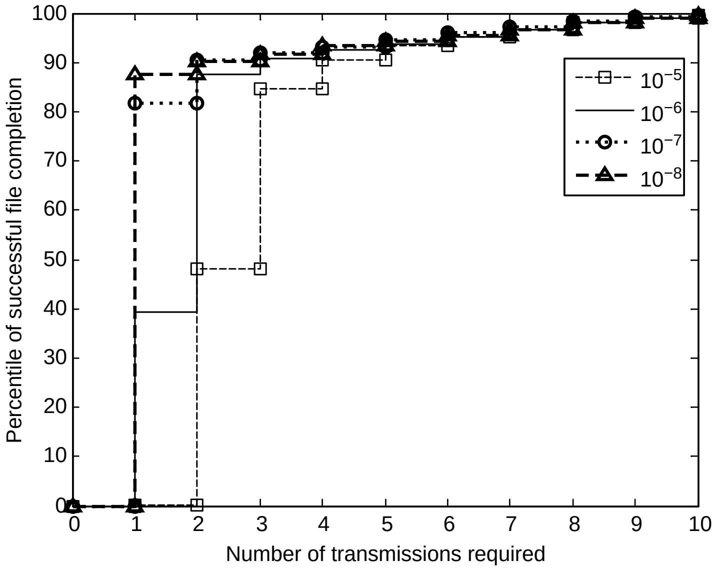


Fig. 8 1 MB file latency performance with bad weather BER at 10^{-4}.

from 0.1 to 1 transmission. The reduction in average latency is quite significant, yielding more than 50% reduction.

Here it should be noted that the file size has more impact on the 99 percentile latency instead of the average latency performance. One possible explanation is that the 99 percentile latency computation has a stronger dependency on the weather correlation at the tail events, whereas the average latency does not reflect

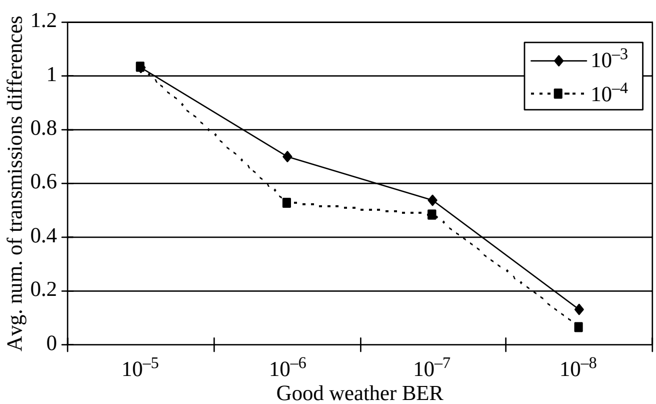


Fig. 9 Average number of transmissions required.

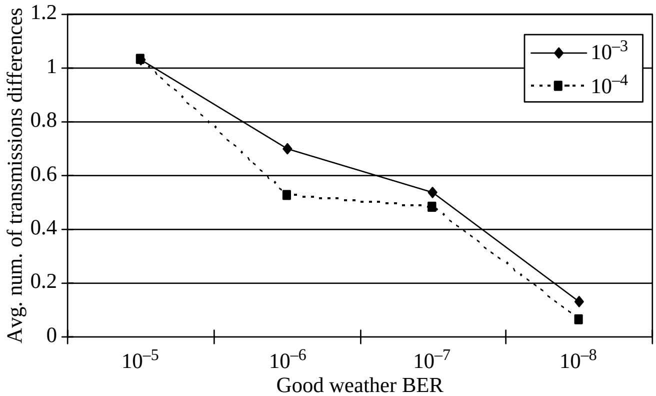


Fig. 10 Differences between the average numbers of transmissions required to achieve 99% of 10 MB and 1 MB file with 10^{-3} and 10^{-4} bad weather BER.

the correlation as strongly because the tail events are averaged out over the entire distribution. Therefore, this indicates that if the system is designed to meet the high percentile latency performance, one needs to improve, on a fundamental level, the Ka-band link's sensitivity to weather events using adaptive methods and weather forecasting as described in [6].

C. Effect on Different Sampling Methods

Data obtained from different sampling methods are compared to evaluate the sensitivity of our analysis. We confirm that choosing the maximum among 28 samples provides the upper bound and choosing the minimum can provide the optimistic lower bound for actual latency. We observe that the three methods of sampling have minimal impact on average file transmission performance, which validates that the result of the 20-minute average sampling is legitimate.

First, computations were conducted using sample data sets obtained from the maximum sampling method. From Figs. 11 and 12, the maximum sampling method yields up to 7 more transmissions for the 99% file completion than the transmissions required from the average sampling method shown in Figs. 4 and 5. The average number of transmissions required to complete 99% of the 10 MB file size is recorded in Fig. 13 with 10^{-3} and 10^{-4} bad weather BERs. On average, about 4.7 transmissions are required to transmit the 10 MB file for the (10^{-5}, 10^{-3}) BER pair and 2 transmissions for the (10^{-8}, 10^{-4}) BER pair.

Also, differences between the maximum sampling method and the average samplings are shown in Table 6. The deviations are around 0.5 or less transmissions between two different sampling methods in terms of predicted average delay performance. Errors due to data alternations from the mis-sampling could be less than the 1 transmission difference. Hence, we can observe that mis-sampling the original data does not introduce significant performance deviations.

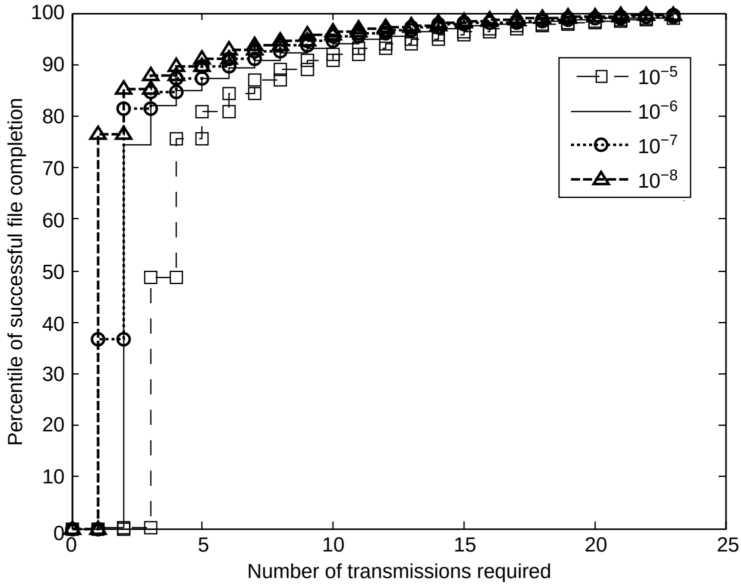


Fig. 11 10 MB file latency performance with bad weather BER at 10^{-3} with data obtained from maximum sampling.

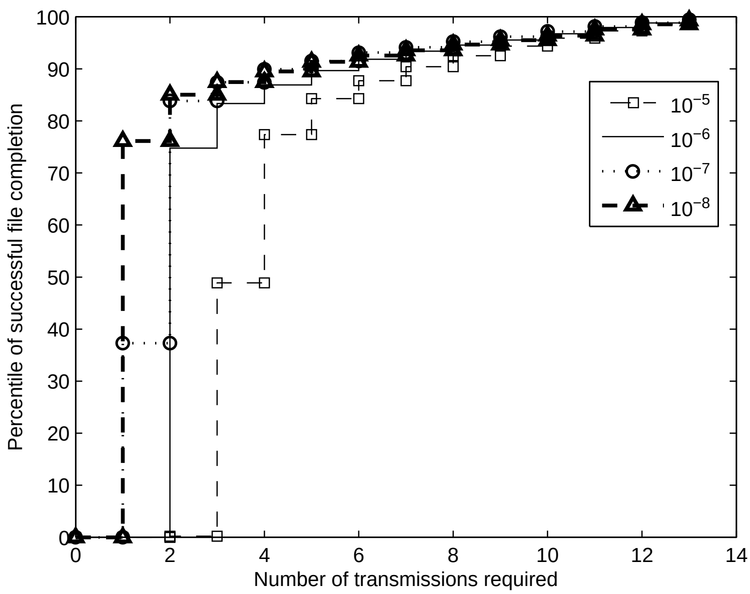


Fig. 12 10 MB file latency performance with bad weather BER at 10^{-4} with data obtained from maximum sampling.

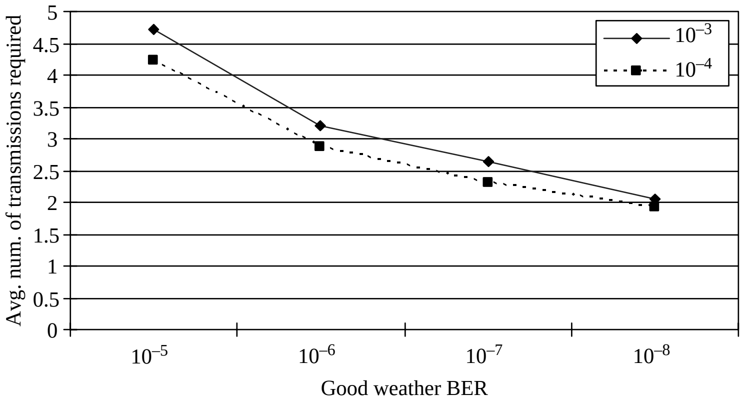


Fig. 13 Average number of transmissions required to achieve 99% completion of the 10 MB file on the data obtained from maximum sampling.

Using the same parameters, the simulation was run with the data obtained from sampling minimum data point. The CDFs of the number of transmissions required for 99% file completion are shown in Figs. 14 and 15 with 10 MB file size.

The minimum sampling method predicted 2–3 fewer transmissions than the result obtained from the average sampling method, as shown in Figs. 14 and 15. More good weather data in the minimum sampling method results the improved delay performance than the results from other sampling methods. The average number of transmissions required to complete the 99% file transmissions are recorded in Fig. 16 for different BER pairs. On average, at most 3.7 transmissions are required to transmit the 10 MB file for the BER value pair (10^{-5}, 10^{-3}), and 1.4 transmissions for the BER pair (10^{-8}, 10^{-4}). Differences between the average number of transmissions required from the average sampling method and the minimum sampling method are recorded in the Table 7.

The overall CDFs of three different sampling methods are compared in Figs. 17–20 for various BER pairs. It is found that the performance of the average sampling method is bounded between the maximum and the minimum sampling method. Because of the high weather correlation, the maximum transmissions requirements to complete 99% of the files are highly affected by the choice of

Table 6 Differences between the average numbers of transmissions required to achieve 99% completion of the 10 MB file from the maximum and the average sampling method

Bad weather BER\ Good weather BER	10^{-5} (good weather)	10^{-6} (good weather)	10^{-7} (good weather)	10^{-8} (good weather)
10^{-3} (bad weather)	0.5115	0.3003	0.4019	0.3301
10^{-4} (bad weather)	0.304	0.3376	0.2898	0.3848

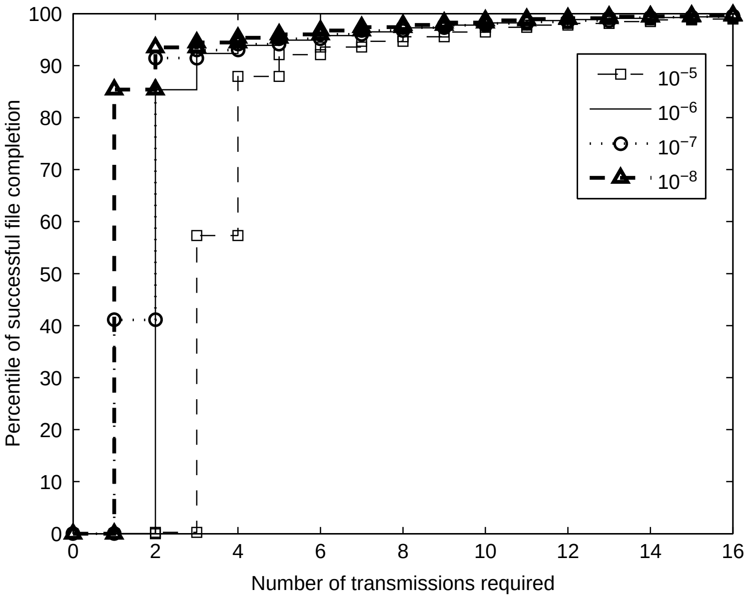


Fig. 14 10 MB file latency performance with bad weather BER at 10^{-3} with data obtained from minimum sampling.

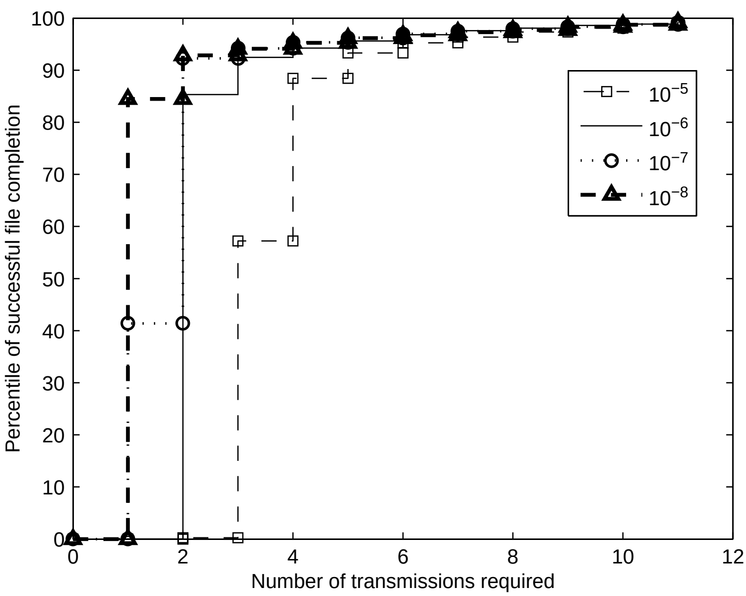


Fig. 15 10 MB file latency performance with bad weather BER at 10^{-4} with data obtained from minimum sampling.

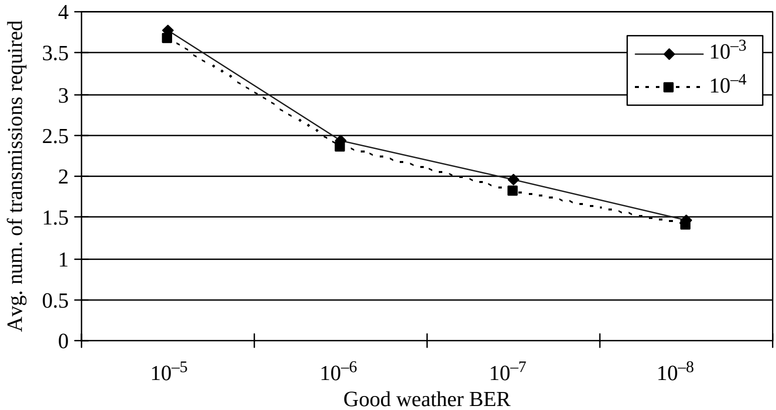


Fig. 16 Average number of transmissions required to achieve 99% completion of the 10 MB file on the data collected from minimum sampling method.

Table 7 Differences between the average numbers of transmissions required to achieve the 99% completion of the 10 MB file with the average sampling and minimum sampling method

Bad weather BER\Good weather BER	10^{-5}	10^{-6}	10^{-7}	10^{-8}
10^{-3}	0.4167	0.4755	0.2685	0.2771
10^{-4}	0.2612	0.1995	0.1922	0.1445

sampling methods. However, they are the upper bounds, and the actual average file transmissions requirements are significantly less.

IX. Conclusion

In this study, we present the framework to evaluate the CFDP performance under good and bad weather conditions. The CDFs of the number of transmissions required to complete the file transfer for both 99 percentile latency and average latency are derived. The BERs at good and bad weather conditions affect the latency performance significantly, and varying file size has a moderate impact on the delay performance. While file size and BER affect the average latency, the 99 percentile latency is more dependent on the correlation of the weather patterns, which demonstrates the importance of developing weather mitigation strategy besides using retransmission protocols. Also, a fundamental improvement in the availability of the Ka-band link is expected to improve not just the average performance but significantly reduce the maximum number of transmissions required in the CDF plot.

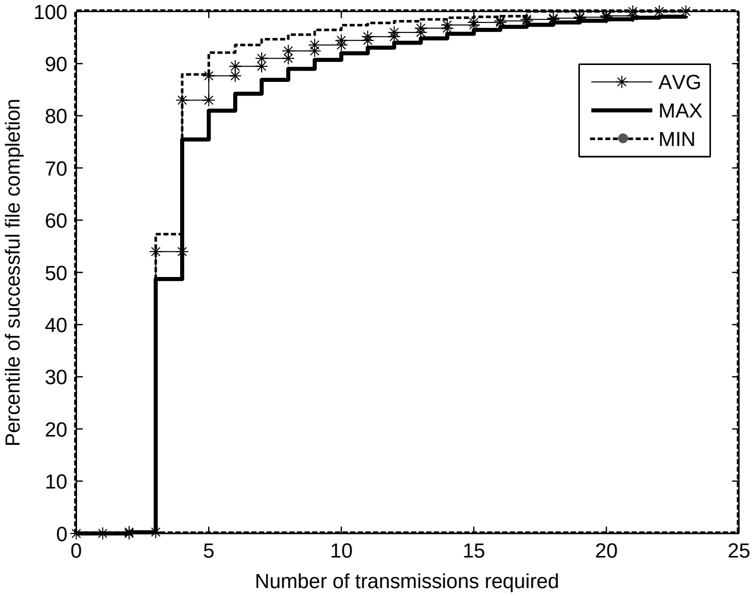


Fig. 17 CDF of 10 MB file latency performance with (10^{-5}, 10^{-3}) BER on data obtained from the averaging, maximum, and minimum sampling method.

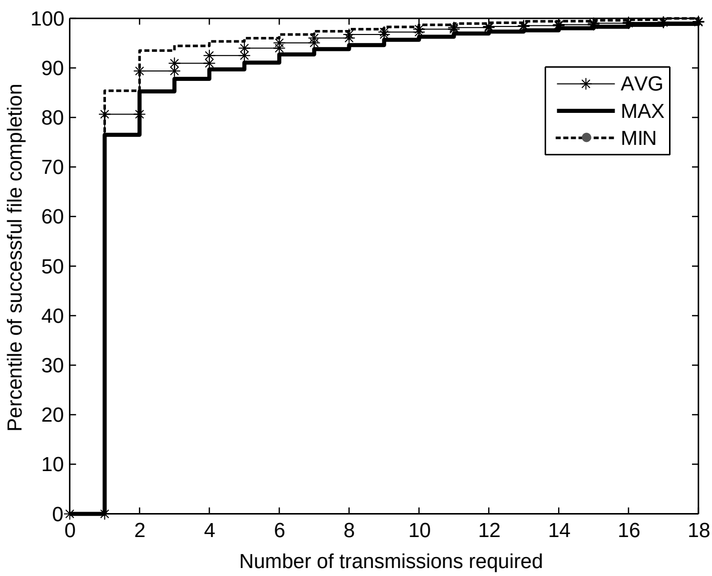


Fig. 18 CDF of 10 MB file latency performance with (10^{-8}, 10^{-3}) BER on data obtained from the averaging, maximum, and minimum sampling method.

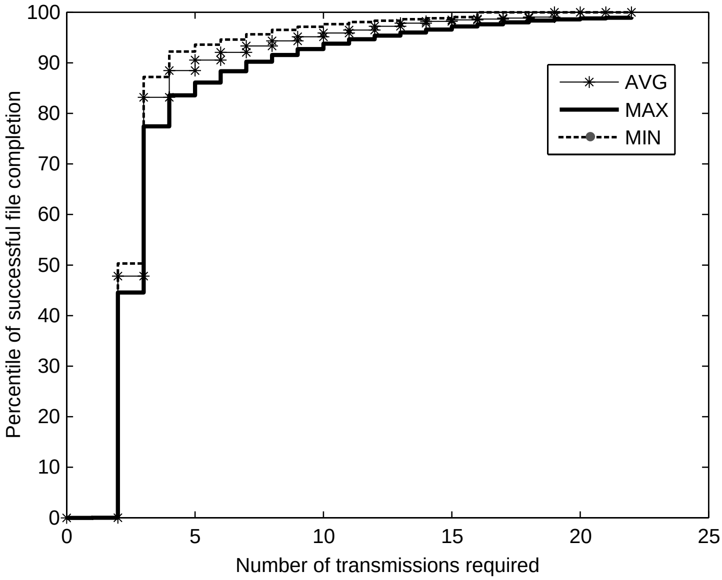


Fig. 19 CDF of 1 MB file latency performance with (10^{-5}, 10^{-3}) BER on data obtained from averaging, maximum, and minimum sampling method.

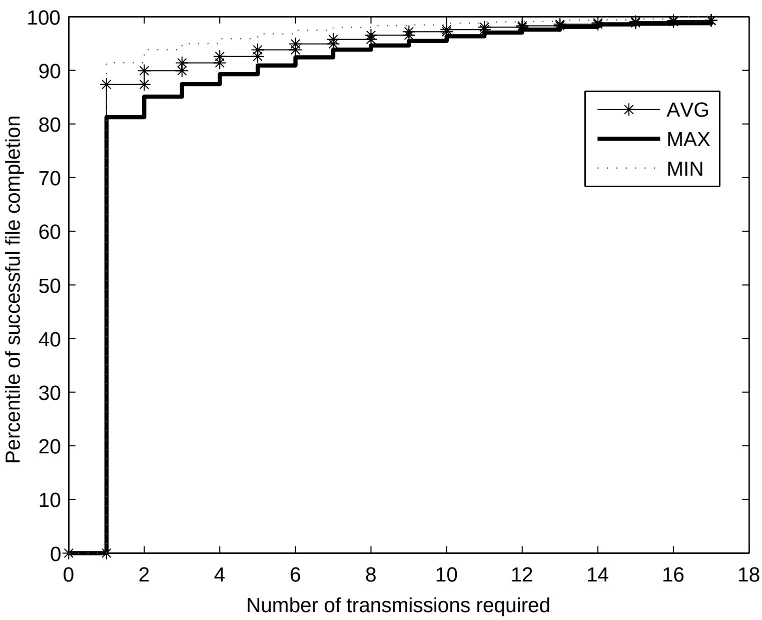


Fig. 20 CDF of 1 MB file latency performance with (10^{-8}, 10^{-3}) BER on data obtained from averaging, maximum, and minimum sampling method.

Acknowledgments

The authors would like to thank Dr. Owhadi at California Institute of Technology for introducing Hoeffding's Inequality and other invaluable comments on Sec. VII. Also, the authors would like to acknowledge fellow researchers in the Communications Research Section at the Jet Propulsion Laboratory for their suggestions and feedback, which greatly clarified and improved this work. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

References

[1] *CCSDS File Delivery Protocol (CFDP)—Part 1: Introduction and Overview,* Consultative Committee for Space Data Systems, CCSDS 720.1-G-2, Washington, DC, Sept. 2003.

[2] Gao, J. L., and Segui, J. S., "Performance Evaluation of the CCSDS File Delivery Protocol – Latency and Storage Requirement," *Proc. IEEE Aerospace Conference,* Big Sky, MT, March 2005.

[3] Li, W., Law, C. L., Dubey, V. K., and Ong, J. T., "Ka-Band Land Mobile Satellite Channel Model Incorporating Weather Effects," *IEEE Communications Letters,* Vol. 5, May 2001.

[4] Chu, V. Y. Y., Sweeney, P., Paffett, J., and Sweeting, M. N., "Characterizing Error Sequences of the Low Earth Orbit Satellite Channel and Optimization with Hybrid-ARQ Schemes," *Proc. IEEE GLOBECOM Conference,* Vol. 5, Nov. 1998.

[5] Jiao, C., Schwiebert, L., and Xu, B., "On Modeling the Packet Error Statistic in Bursty Channels," *Proc. 27th IEEE Local Computer Networks Conference,* Nov. 2002, pp. 534–541.

[6] Sun, J., Gao, J., Shambayatti, S., and Modiano, E., "Ka-Band Link Optimization with Rate Adaptation," *Proc. IEEE Aerospace Conference*, March 2006.

[7] Feyerherm, A., and Bark, L., "Goodness of Fit of a Markov Chain Model for Sequences of Wet and Dry Days," *Journal of Applied Meteorology,* Vol. 6, No. 5, 1967, pp. 770–773.

[8] Zorzi, M., Rao, R., and Milstein, L. B., "On the Accuracy of a First-Order Markov Model for Data Transmission on Fading Channels," *Proc. IEEE International Conference on Universal Personal Communications,* Nov. 1995, pp. 211–215.

[9] Gilbert, E. N., "Capacity of a Burst-Noise Channel," *Bell System Technology Journal,* Vol. 39, Monograph 3683, Sept. 1960, pp. 1253–1266.

[10] Garcia, L., *Probability and Random Processes for Electrical Engineering,* 2nd ed., Addison Wesley, Reading, MA, 1993.

[11] Elbert, B., *Introduction to Satellite Communications,* 2nd ed., Artech House Inc., Boston, MA, 1999.

[12] Hoeffding, W., "Probability Inequalities for Sums of Bounded Random Variables," *American Statistical Association Journal,* Vol. 58, No. 301, March 1960, pp. 13–30.

III. Ground Systems

This page intentionally left blank

Chapter 9

Development of JWST's Ground Systems Using an Open Adaptable Architecture

Bonnie Seaton,* Alan Johns,† and Jonathan Gal-Edd‡
NASA Goddard Space Flight Center, Greenbelt, Maryland
Curtis Fatig§
Science Applications International Corporation, Greenbelt, Maryland
Ronald Jones¶
ASRC Aerospace Corporation, Greenbelt, Maryland
and
Francis Wasiak¶
General Dynamics Corporation, Greenbelt, Maryland

I. Introduction

THE James Webb Space Telescope (JWST) is a large aperture infrared space telescope with a five-year mission, 10-year design goal. It is currently planned to be launched in 2013 from Kourou, French Guiana, aboard an Ariane 5 launch vehicle. JWST is designated to succeed the Hubble Space Telescope (HST) as part of the National Aeronautics and Space Administration (NASA) Great Observatories program. JWST will continue the HST tradition of advancing breakthroughs in our understanding of the origins of the earliest stars, galaxies, and the very elements that are the foundations of life.

---

*JWST Science Operations Manager.
†JWST Ground Segment Manager.
‡JWST Senior System Engineer.
§Senior System Engineer, IGSS Development.
¶Senior System Engineer, JWST Systems.

Fig. 1 JWST team.

JWST, with development and operations phases that may each exceed 10 years, requires non-traditional means of defining the development, and the integration and test (IT) environments and operational ground system. Rather than selecting the real-time control system and building the other ground system components around it, JWST has elected to develop requirements for the desired operational architecture and then select the products necessary to build the development and IT architectures, using the operational architecture as a template.

The JWST team, partially shown in Fig. 1, includes several partners at multiple locations: 1) project management located at Goddard Space Flight Center (GSFC), 2) observatory prime contactor [Northrop Grumman Space Technologies (NGST)], 3) Integrated Science Instrument Module (ISIM) located at GSFC, 4) Near-Infrared Camera (NIRCam) provided by a team led by the University of Arizona, 5) Near-Infrared Multi-Object Spectrometer (NIRSpec) provided by the European Space Agency, 6) Mid-Infrared Instrument (MIRI) provided by an international collaboration led by the Jet Propulsion Laboratory, 7) Flight Guidance System (FGS) provided by the Canadian Space Agency, and 8) Science and Operations Center located at the Space Telescope Science Institute (STScI) located in Baltimore, Maryland.

II. Evolution of the Ground System

From a ground system perspective, the JWST mission system is a fairly traditional science mission. For operations, the JWST will be located at the second sun-Earth Lagrange point (L2, see Fig. 2) that provides JWST with a naturally cold environment with no solar eclipses and an unobstructed view of the Earth for communications. After launch and commissioning, normal operation will involve a single 4-h contact per day to allow the uploading of commands, monitoring of real-time engineering telemetry, and downlinking of recorded engineering and science data. The ground processing of the recorded science data is done overnight so that the products are available to the users the next day. The engineering data

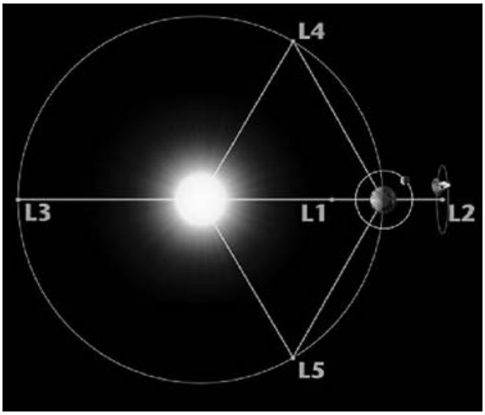


Fig. 2 Lagrange points.

are dumped first and sent to the analysis system in near real-time to determine any backorbit out-of-limit conditions.

JWST has a project requirement to use the same ground system for all phases of the mission: science instrument and spacecraft development, IT and mission operations. The intent of this requirement is to follow a "test-as-you-fly" philosophy, to identify problems very early in the project life cycle and to reduce risk during operations. To satisfy the requirement, an initial determination was made as to the scope of the ground system for operations and the subset that will be used for the IT system. The consideration of the various ground system choices had to provide not only the best technical solution but also the best business solution that will be acceptable with the prime contractor and various international partners. This meant using a commercial, off-the-shelf (COTS) product rather than a specially developed product.

The IT of the JWST will occur in a distributed fashion with the assembly of the observatory occurring at the prime contractor's facility in Redondo Beach, California, cryogenic testing taking place at Johnson Space Center (JSC), and launching from French Guiana. To use the same ground system in all of these locations requires a system that is easy to set up and network, secure and power using U.S. and international connectors.

Providing an IT system 10 years prior to launch created an opportunity to design upgrade paths into the ground system and take advantage of upcoming technologies. Most missions have a short window (less than five years) between the time development starts and launch and have to select a command and telemetry system (usually the one the mission team is "used to") and tailor the mission operations concepts around it. JWST decided to take a different approach and address "new" concepts for mission operations, by having modular plug and play components. This is a departure from selecting a real-time system and building the functionality around it.

The JWST system includes the systems typically used for most science missions, starting at the science investigator proposal system and ending with the science

data system delivering science products to the science investigator. The earlier development and IT ground systems are based on the final operational system. The overall high-level ground system is depicted in Fig. 3. The main components of the JWST ground system are:

1) Science Proposal Planning System (PPS). Provides the proposal solicitation, processing, and planning functions required to implement the science program and to generate the observation plan.

2) Observatory Scripting System (OSS). Provides the tools necessary to assist in the development, validation, and management of onboard scripts.

3) Flight Operations System (FOS). Provides the command data uplink and telemetry capture functions, performs telemetry processing necessary to monitor observatory status, monitors observatory and ground status, and notifies operations personnel in the event of an anomaly detection.

4) Project Reference Database System (PRDS). Comprised of the project reference database as well as the necessary tools to manage the inherent data. It is the repository for all JWST data and information required for observatory operations such as telemetry descriptors, commands, parameters, algorithms, and characteristics. It provides for the configuration management, change process management, and data distribution functions required to provide operational data to the other components of the ground system.

5) Data Management System (DMS)/Science Archive. Provides the data processing, archive, catalog, calibration, distribution, and analysis functions required to support the science program and maintenance of observatory performance. One of the primary components of this system is the science pipeline. The

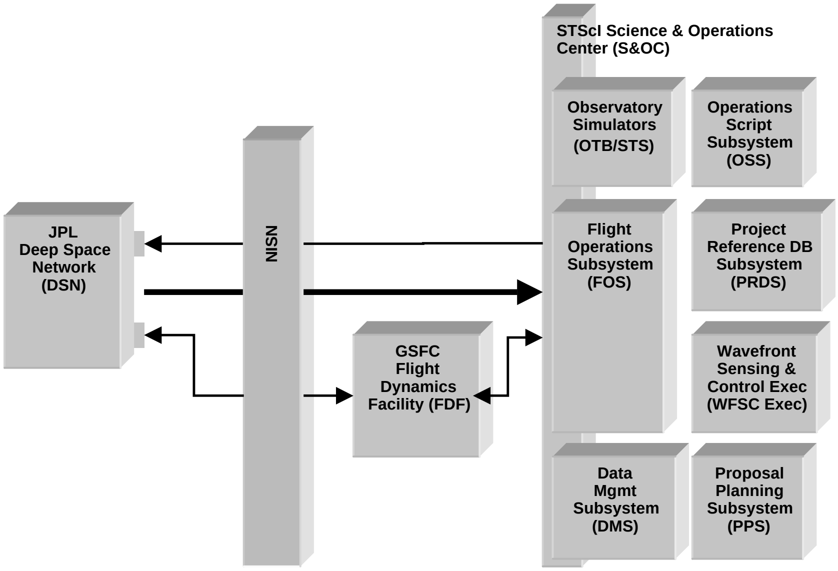


Fig. 3 High-level ground system overview.

science pipeline provides an automated means of low-level processing of the science data and associated engineering data.

6) Wave Front Sensing and Control (WFSC) Executive. Stores and processes science and engineering data obtained to measure wavefront error. Produces commands that will be uplinked to correct the optical figure of the telescope.

7) Flight Dynamics Facility (FDF). Provides the mission's orbit determination, tracking, and ranging support.

8) Deep Space Network (DSN). Provides the flight-to-ground communication portion of the mission.

9) NASA Integrated Services Network (NISN). Provides the telecommunication services needed for the transmission of data among the other ground elements.

A few of the key features that will make the JWST ground system modular are 1) use of eXtensible Markup Language (XML) for the PRD, 2) use of web-based technologies for the user interfaces, and 3) use of a generic format for the engineering data, thus separating the real-time system engineering format from engineering archive format. All of these features will be discussed in the following sections.

III. Implementation of the Ground System

The initial version of the development and IT versions of the ground system were built around the FOS and the PRDS, along with the science processing pipeline portion of the DMS. The science pipeline provides a seamless means of taking the raw science files and processing them into the end user's format, Flexible Image Transport System (FITS). During the later phases of IT, the OSS, the WFSC Executive and parts of the PPS are also integrated into the IT version of the ground system. Each part of the various components is broken down to the least logical unit.

Part of the idea of using as much COTS as possible was to benefit from future product enhancements provided by the vendor. For example, it is expected that future versions of the real-time system will include a robust web interface.

The transportable items are 1) command definitions, 2) telemetry definitions, 3) engineering trending and data archive, 4) interfaces between the command/control system and the end unit, and 5) science processing.

The logical units needed for the first systems to be used during the development and IT of the science instruments and flight software containing the components from the eventual operations center are a real-time command/control system, a database system, and NISN-type interface. The team also realized the need to develop an independent database tool based on an XML database system, rather than use a tool provided by the real-time command and telemetry system vendor. This approached provided benefits right away as various real-time command and telemetry systems are used in early development. The first system (circa 2001) for the GFSC flight software (FSW) development effort contained an XML database system, EPOCH® real-time ground system, and a GSFC-provided data formatter (DF) for the NISN interfaces. In 2002, the real-time ground system was changed to ASIST® with only minor impacts. By defining the interface control document (ICD) between the real-time ground system and the end unit and by using XML for the database input format, the development time

for the software to translate from the XML to the ASIST® format was one week. The next change occurred in 2003 when the real-time ground system was changed to Eclipse® to be compatible with the prime contractor's IT system and the future operational real-time command/control system. The Eclipse® transition took about one month mainly due to the unique ground header requirements. By being open and adaptable, the current DF and PRD support both the Eclipse® and ASIST® real-time ground systems.

In 2004, JWST expanded the deployment of the ground system to development sites outside of GSFC. The deployment of the first integration and test ground systems, referred to as Science Instrument Development Units (SIDUs), has been completed. Eighteen systems were built and have been deployed to GSFC; JPL; Palo Alto, California; Ottawa, Canada; Cambridge, Canada; Munich, Germany; Madrid, Spain; and Abingdon, England. The SIDU is shown in Fig. 4, and is used by the various groups to develop JWST flight software.

The next evolution of the JWST ground system is the Science Integration Test System (SITS) shown in Fig. 5. The SITS will be used to perform integration and testing of the hardware components including flight hardware. Deployment of the seven SITS is currently under way, with the first deployment to the MIRI facility in Abingdon, England, occurring in April 2006. The remaining deployments will occur in 2006 and 2007.

The final two integration and test ground systems will be the Instrument Test Support Systems (ITSS) that will be used for cryogenic and final integration testing.

The 27 SIDU, SITS, and ITSS ground systems must be transportable, integrate into a facilities network, and operate on an isolated network. All of these test

Fig. 4 JWST SIDU.

Fig. 5 JWST SITS.

systems must be compatible with one another as well as the prime contractor's test systems and the operational system that will reside at STScI.

IV. Ground System Implementation Decisions

To deal with a project of this scope where technology and software will be continually evolving, a core set of implementation decisions has been established and is listed next. These implementation decisions are factored into the JWST ground system core components and drive the overall design of the JWST ground system. To date, these decisions have proven very successful as systems are developed. The core set of implementation decisions include the following:

1) Use the operational telemetry and command system for integration and test:
 a) Use the same data and interfaces throughout the life of JWST.
 b) Design modular components at the start of the development.
 c) Provide upgrade path from the beginning of the process.
 d) Explore automation technologies such as system messaging.

2) eXtensible Markup Language (XML):
 a) JWST XML compatible with CCSDS XML Telemetric and Command Exchange (XTCE).
 b) Database is just the data, not tied to a particular application.
 c) Allows for cross-referencing of command, telemetry, and operations products.
 d) Engineering data saved in a manner to be application independent for data analysis.

3) Project reference database:
 a) Common area for all mission-related information for the real-time system, planning system, and spacecraft characteristics.
 b) Data independent of any system.
 c) Certification and configuration management of mission-related information.

4) Onboard scripts:
 a) Advantage of increased processing power of the PowerPC flight processor for event-driven operations.

b) Use of JAVA Script COTS engine.
c) Use of modular and common components onboard.
d) Data dictionary.

5) Consultative Committee for Space Data Systems (CCSDS) File Delivery Protocol (CFDP):
a) Reduce functionality needed at control center, yet increase data reliability by providing a reliable file downlink protocol.
b) Use CCSDS standards for software and maintenance.
c) Use Deep Space Network to provide level-0 processing of science data.

6) Batch decommutated data:
a) Common generic format for all engineering data.
b) Engineering exchange format for other ground system components.
c) Data storage format prior to ingest into the data warehouse.

7) Engineering archive and trending:
a) Common engineering data store for the life of the mission.
b) Provide automated reporting.
c) Provide tools to analyze the status and performance of the JWST observatory.

The operational telemetry and command system that is also used for the integration and test phase of JWST is covered by [1].

XML was chosen to provide the structure for the JWST database. The JWST database and CCSDS XTCE standard interrelationship is described in [2, 3]. The JWST XML has been compared with the Jet Propulsion Laboratory's (JPL) XML as an independent study on the commonality between XML databases. It was realized after a few short technical meetings that the JWST database could be a subset of the JPL XML database and that 90% of the two databases matched. JWST also has been working with the CCSDS XTCE committee on developing its standards. These efforts gave JWST confidence that all of the necessary items have been covered and the structure of the JWST database was reasonable and conformed with other XML databases.

PRD is the central hub of the ground system, yet the ground system was not designed around it. The PRD continues to be modified as necessary but because of the nature of XML, it remains backwards compatible. JWST has also extended the XML database with XML metadata to cover those items that are not XML compatible, such as display pages, scripts, loads, etc, so that they can be included in the JWST database and be compatible with all of the JWST database tools. JWST has developed database tools and style sheets for the users to enter in database items. A majority of the database rules are checked as the user enters in the data either at the local laboratory or the PRD central site, greatly reducing the number of errors. The ability to quickly create a database for the real-time IT system allows the users to check (and correct as needed) the database locally before submitting inputs to the official central database. Also by building a method for quick database creation, the workarounds usually done in the IT systems to update databases are eliminated.

Onboard scripts are Java scripts in human-readable format that will be uplinked to JWST as a file. These scripts will configure the spacecraft and instruments to perform science operations and provides for the implementation of event-driven operations. Useful features of scripting include that the failure of a script will not

cause any spacecraft problems, a user can read exactly what is happening onboard without special tools, and the script can be easily executed on the ground using the same software engine that resides onboard.

A major concept in designing the JWST ground architecture is separating the real-time telemetry stream from recorded streams. The real-time streams are supported using the standard ground system, DSN and NISN interfaces. File processing allows the DSN ground station to perform level-zero processing on recorded data and store the files locally at JPL until the mission operations center is ready to retrieve it. The files are transmitted using a file transfer protocol to the JWST mission operations center.

For JWST, the CCSDS CFDP protocol provides three functions: file load, file dump, and SSR dump capabilities. For file loads, the CCSDS CFDP provides a means for identifying source and destination file naming, added verification (in addition to the COP-1 verification) that the file was uplinked with no errors, onboard directory placement of the files, cancellation and overwriting of files as necessary. The file dumps using CCSDS CFDP verification guarantee that the file was received correctly with no errors with knowledge of the source and destination file naming. For the downlinking of recorded data, CCSDS CFDP provides the greatest benefit; guaranteed delivery of the data to the ground, onboard automatic releasing of SSR space as portions of the dump are verified, and level zero processing are done at the ground station.

Batch decommutated data are in a generic engineering format used in IT and the eventual real-time operations system. A JWST front-end system ingests CCSDS packets and outputs the generic engineering format that is defined in a JWST ICD. The purpose of this generic engineering format is to allow other systems to be compatible without having to deal with COTS proprietary data formats.

Engineering archive and trending, also known as the engineering data warehouse, is not planned for the JWST IT ground system architecture. It is planned in the future operational system and will include JWST IT data collected during critical testing. The input format for the engineering archive and trending system will be compatible with the batch decommutated data structure that is used during the JWST IT. The current plan for real-time operations is for the batch decommutated data to receive the recorded and real-time engineering data. After the engineering data are put into the generic format, the engineering archive and trending system will receive a bulk load of the data. The users will be able to receive the data in various formats, plots, reports, etc., as well as set up standard reports to be automatically executed on a time or event interval.

V. Success and Failures of Adaptability

So far, the successes of the JWST open adaptable architecture have outweighed any setbacks. In retrospect, two items that could have been handled differently include:

1) Use the same command and telemetry system earlier than implemented in the flight software development facility.

2) Process the paperwork for the Technical Assistance Agreements (TAA) and International Traffic and Arms Regulations (ITAR) earlier in the development process.

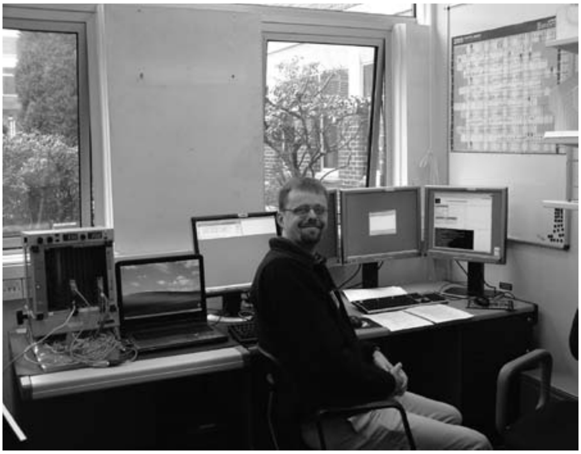


Fig. 6 Satisfied user.

The successes have been the following:

1) Interoperability has been implemented with three TC systems and two database systems. Two trending systems, two front-end systems, and a planning system have also been prototyped over the past three years.

2) Central and local database systems using XML have allowed JWST to build databases in minutes and to be compatible with other mission databases as well as with CCSDS XTCE.

3) Adaptability has been achieved with the deployment of five systems internationally. One system is shown in Fig. 6. All work in their local environments and interface with the central database. New functionality and maintenance updates have also been provided remotely.

4) A help-desk, Web-based problem reporting system informs users of any problems plus allows users to submit a problem report at any time.

Development of the key elements of the JWST ground system used for development, IT, and operations followed an evolutionary process for taking one step at a time, not trying to do too much at one time. The spacecraft is shown in Fig. 7. Steps that JWST is using to introduce technologies into the overall ground system are 1) separate the database from any particular system or application, 2) batch decommutation of engineering data into a generic format, and 3) use Web-based technologies for end user displays such as real-time pages, plots, problem reporting, archive retrievals, and reports.

VI. Conclusion

After four years of real-life experiences with an open architecture, the JWST ground system team has noted the following lessons learned:

1) Central XML database that is application independent was instrumental in minimizing the cost and impact as the database matures. The JWST database

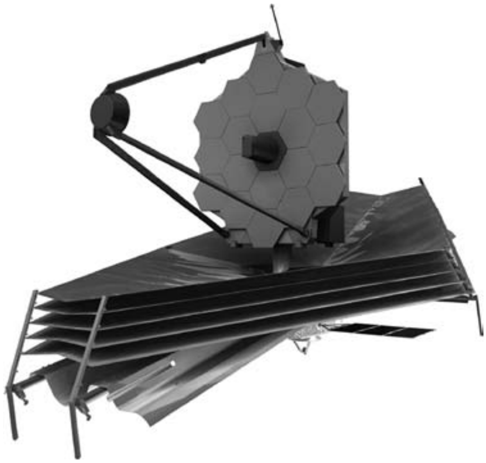


Fig. 7 JWST.

XML implementation also allowed for JWST to be compliant with the CCSDS XTCE without much effort.

2) Various vendors supplying different components works well for keeping the total system open and adaptable, but increases the amount of coordination between the various delivery schedules.

3) Separating real-time functions from engineering functions is a major step for ground systems.

4) Interoperable lug and play concept works as long as the ICDs are defined for the interfaces. Also the middleware software for exchanging information between systems needs to be thought about at the beginning of system development, even though implementation may be years away.

5) Open engineering and science data formats that are defined in an ICD allow for dissimilar systems to have access to the data without impacting the design.

6) CCSDS CFDP reduces the amount of processing needed at the end user site, increases the data efficiency, and eases the problems of onboard recorder management. This CCSDS protocol is a win-win for all users.

7) Use of web-based technologies for the end user displays provides more flexibility, quicker development, and reduces long-term cost.

References

[1] Woods, R., and Burns, W., "The James Webb Space Telescope Common C&T System: Interoperable by Design," AIAA Paper 2006-5687, Rome, June 2006.

[2] Gal-Edd, J., Merri, M., and Oyake, A., "Exchanging Databases with Dissimilar Systems Using CCSDS XTCE," AIAA Paper 2006-5801, Rome, June 2006.

[3] Merri, M., "CCSDS Standardization in Spacecraft Monitoring and Control," AIAA Paper 2006-5756, Rome, June 2006.

This page intentionally left blank

Chapter 10

Ground Data System Services: A Service-Oriented Architecture for Mission Operations

Roger S. Thompson,* Stewart Hall,† and Oliver Page‡
SciSys, Ltd., Chippenham, England, United Kingdom
and
Nestor Peccia§
European Space Agency, Darmstadt, Germany

I. Introduction

THE objective of a service-oriented architecture (SOA) is to achieve loose coupling between interacting software components or *agents*. In traditional software systems, where components are strongly coupled, the interface between any pair of subsystems is often explicitly defined in terms of the full stack of protocols employed from application down to low-level communications. This leads to closed architectures, in which it is difficult to replace, or reuse, software components from one system to another (see Fig. 1).

In SOA, a backbone of standardized service interfaces is defined in which individual components act either as a provider or consumer of the service. Providers (or servers) offer their capabilities through the published service interfaces, while the consumers (or clients) use those services. It should be noted that the direction of data flow does not define which component is provider and which is consumer—data can flow in both directions. Application-level components only communicate with each other through these standard services and may be "plugged in" to a supporting infrastructure that implements the service interfaces (see Fig. 2). Services are also layered, such that specific application-level services are implemented using more generic messaging services, which themselves use an underlying communications protocol.

---

*Chief Architect, Space Sector.
†Consultant, Space Sector.
‡Senior Applications Developer.
§Head of Data Systems Infrastructure, European Space Operations Centre.

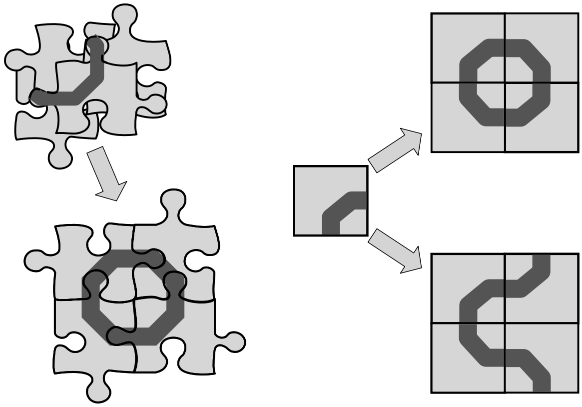


Fig. 1 Traditional vs service-oriented architecture. The components of a traditional architecture can be integrated into just one system. With service-oriented architecture, many similar systems can be deployed without modification to the components.

Specification of a standard service interface requires 1) an information model for the service, which defines the objects exposed at the service interface that are meaningful in the application domain, and 2) a dynamic model that defines both the operations that a service consumer can invoke the provider to perform on those objects and the events that the service provider uses to report changes in the state of those objects to consumers. This approach unifies the definition of messages exchanged between service provider and consumer, with the associated configuration (operations preparation) data and logging of history.

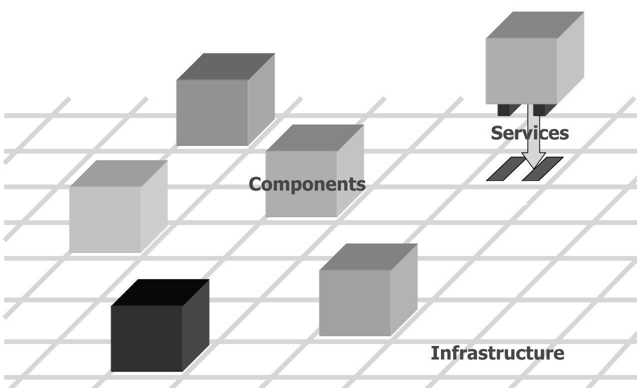


Fig. 2 Service-oriented architecture: plug-in components, services, and infrastructure. Components only communicate with each other via standardized service interfaces. These services are themselves implemented over a distributed communications infrastructure. Both plug-in components and the infrastructure itself can be replaced without change to the rest of the system.

The potential benefits of the SOA approach, as applied to spacecraft mission operations, include:

1) Plug-and-play interoperability of mission control system (MCS) components.
2) Reuse of common infrastructure across multiple systems.
3) Independence of core application software from underlying implementation technology—platform and communications.
4) Scope to evolve a system, by replacing components or changing underlying technologies.
5) Reduced mission-specific deployment costs.
6) Independence of mission configuration data and history from system implementation.

II. Ground Data System Services

ESA is currently developing the architecture for the future ESA Ground Operations System (EGOS) and a number of studies are supporting this. The Ground Data System Services (GDSS) study [1], performed by SciSys, has specifically focused on the definition of end-to-end mission operations services. This study draws on the approach developed in the context of the Consultative Committee for Space Data Systems (CCSDS) Spacecraft M&C Working Group as outlined in their Green Book, *Mission Operations Services Concept* [2].

A. Mission Operations Functions

The ground segment has been decomposed into systems in the European Cooperation for Space Standardization (ECSS) ECSS-E70 [3] and modifications to this have been proposed in the European GS Software Technology Harmonization Reference Architecture. From the perspective of mission operations, the ground segment may be considered to comprise the following systems:

1) Ground Station Network (GSTS).
2) Mission Control (Operations) System (MCS).
3) Mission Exploitation System* (MES).
4) Ground Support System (GSUS).

Figure 3 illustrates the principal application-level functions associated with mission operations and their end-to-end interactions.

Functions are grouped into the four principal ground segment systems and, as mission operations are concerned with operation of the space segment, the spacecraft itself. Functions shown with a person symbol typically require a man-machine interface. The lines between functions represent the end-to-end interactions involving the mission operations (MCS) functions that GDSS is principally concerned with. It is these interactions that are to be supported by a set of service specifications specific to the mission operations domain: the *mission operations services*.

The principal MCS application-level functions identified are the following:

*ECSS-E70 decomposes the MCS into functionally equivalent Operations Control System (OCS) and Payload Control System (PCS), and functionally distinct MES. For the purposes of this discussion, MES is therefore considered distinct from MCS.

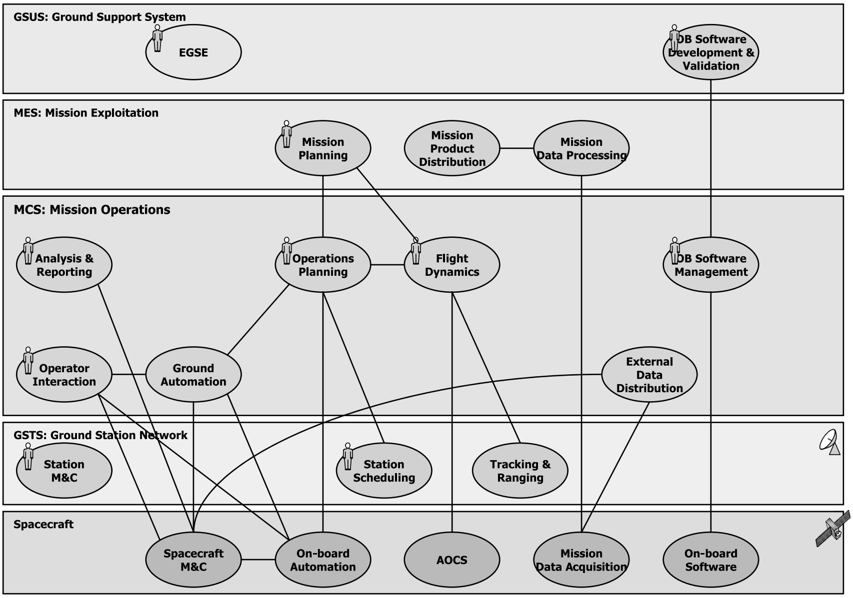


Fig. 3 Principle end-to-end mission operations functions.

1) Operations (mission) planning.
2) Flight dynamics.
3) (Performance) analysis (or evaluation) and reporting.
4) Onboard software management.
5) External data distribution.
6) Operator interaction (status displays and manual control applications).
7) (Ground-based) operations automation.

It is important to note that only those functions constituting the applications layer for mission operations are shown. Other functions whose purpose is essentially to support or implement the application-level interactions are omitted. This includes telemetry/telecommand (TM/TC) protocol-level processing, internal data distribution, and data archiving.

The first four may be considered off-line functions, while the last two concern on-line operations. These MCS functions support end-to-end interaction with applications residing 1) on the spacecraft, 2) at the ground station, or ground station network complex, 3) within the MES or an external ground system, and 4) within the ground support system.

B. Mission Operations Information

The previous section summarizes the mission operations functions and the end-to-end interactions that are to be supported by mission operations services. This is a functional view of the system. To ensure commonality and reuse, it should not, however, be concluded that there is a one-to-one correspondence between the

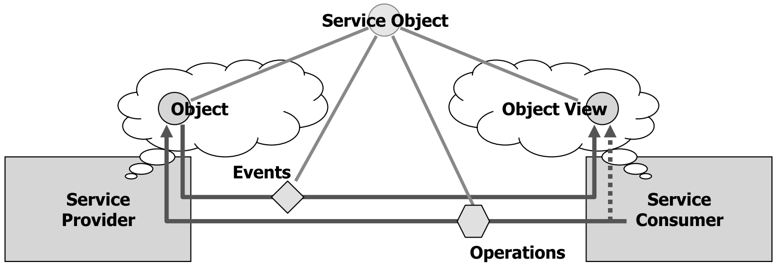


Fig. 4 Information view: service objects, events, and operations.

interfaces between functions (the lines connecting functions in Fig. 3) and services. A service relates to a particular type of interaction, while the interface covers all interactions between any two functions.

To move from the identification of interfaces to the identification of *services*, it is necessary to consider the nature of the information that flows across the interfaces, and what operations can be invoked across those interfaces. This is an information view of the system. When this is analyzed, it is found that:

1) The same fundamental type of information flows across multiple interfaces.
2) An interface can involve several different fundamental types of information.

In fact, the interactions between functions can be reduced to a relatively small set of these fundamental types of information object, which include:

1) Monitoring and control: parameters, actions and alerts.
2) Automated procedures or functions, schedules and planning requests.
3) Time, position, orbit, and attitude.
4) Onboard software images.
5) (Payload) data products and reports.
6) Operator interactions (notifications, alarms, and queries).
7) Data buffers.

These fundamental types of information correspond to information or *service objects* that must be shared by both service provider and service consumer. These service objects are "exposed" at the service interfaces, i.e., their state is known to both consumer and provider. To keep their internal views of a service object in step with each other, consumer and provider must exchange messages (Fig. 4).

Typically, the service provider will generate event messages to signal a change of state in a service object, while a service consumer may invoke operations to effect a change in a service object.

The mission operations services identified correspond closely to the fundamental service objects that have been identified in the information view.

C. Identification of Mission Operations Services

Figure 5 shows the functions previously identified in Fig. 3, linked by colored lines representing the services. Each service is shown in the manner of a bus—the horizontal lines representing the service with consumer functions connected by vertical lines. Provider functions are indicated by the circle on the line connecting

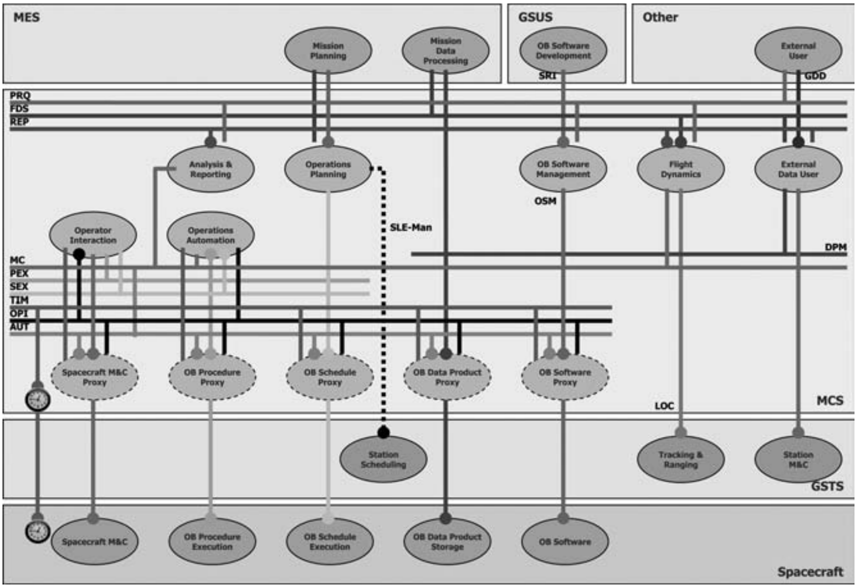


Fig. 5 GDSS mission operations services. (See also the color figure section starting on p. 645.)

them to the service bus. Services providing an interface between just two functions are shown as a simple vertical line.

The identified mission operations services are also listed and described in Table 1.

The services themselves have been identified on the basis of common types of information or control exchanged between the functions. It should be noted that the interface between any pair of functions may be supported by multiple services.

Another key feature shown in the diagram is the concept of proxy functions that represent onboard functions within the ground segment. Proxy functions serve a dual purpose:

1) They can be permanently available in the case of intermittent contact with the space segment. This allows them to hold an image of the last known or predicted status of a spacecraft out of contact; to buffer messages to be sent to the spacecraft; and to act as the repository for service history (archived data) within the ground segment.

2) They can encapsulate a legacy, lower communications protocol-level or nonstandard interface with the spacecraft.

In the context of existing ESA MCS infrastructure, the bulk of SCOS-2000 functionality (other than its user interfaces) is essentially that of the spacecraft proxy.

It should be noted that while the distribution of functions shown in Fig. 5 is representative, it does not reflect all combinations of functional deployment and service usage possible within the architectural framework. Functions such as

Table 1 GDSS mission operations services

Identification	Name	Description
MC	Core monitoring and control	Parameters: publish status; set Actions (Commands): publish status; invoke/send Alerts (Events): notify; raise
AUT	Automation	Specialization of MC for automation of proxy functions
DPM	Data product management	Data product (payload data file): directory; transfer
FDS	Flight dynamics	Orbit/attitude: determination, propagation, maneuver preparation
GDD	Generic data distribution	Product: catalogue; order; deliver
LOC	Location	Position: tracking, ranging, onboard positioning
OPI	Operator interaction	Message/alarm/query: notify; operator response
OSM	OB software management	Onboard software: load; dump
PEX	Procedure execution	Procedure/function: control; progress reporting
PRQ	Planning request	Planning request: request; response
RBM	Remote buffer management	Buffer: catalogue; retrieve; clear
REP	Report	Reports: publish; catalogue; retrieve; generate
SEX	Schedule execution	Schedule: distribute; edit; control; progress reporting
SRI	Software reference image	Onboard software image/patch: distribute
TIM	Time	Time: report; set; correlate; notify

mission planning or flight dynamics could be migrated onboard and associated proxy functions added. Functions can also interact with additional services, e.g., automated flight dynamics could support the schedule execution service.

III. GDSS Service Structure

A. Generic Structure

All GDSS application-level or mission operations services share a common service structure. This is illustrated in Fig. 6. People often equate the service interface to the "live" exchange of information between service provider and service consumer. However, this is only part of the infrastructure required to support the service interface, termed the active service interface in Fig. 6. It is a key concept within GDSS that all aspects of the information flowing across the service interface are integrated within the service layer. Other service level interfaces are required for the following:

1) Service location via the service directory.
2) Distribution and access to the service configuration data.

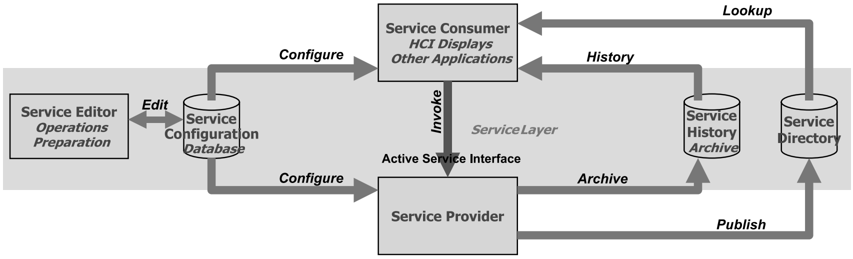


Fig. 6 Generic GDSS mission operations service structure.

3) Archiving and retrieval of the service history.

The service layer involves six main elements:

1) The service provider is responsible for supporting the core service functions.

2) The service consumer is a user of the core service functions, and is typically either a human-computer interface, or another software application.

3) The service directory holds details of all available services. Service providers publish the services they provide within the service directory. Service consumers can then look up a required service in the service directory to locate it, before invoking the service directly with the service provider.

4) The service configuration data specifies the objects and operations that can exist across a specific deployment (or instance) of the service interface, and must be available to both service provider and service consumer if they are to communicate effectively.

5) The service editor enables configuration of the service configuration data for a given instance of the service.

6) The service history (typically held in an archive) constitutes persistent storage of service events, such that a service consumer can retrieve historical status information pertaining to the service.

The service specification does not define the actual implementation of the service provider or consumer. It confines itself to definition of the GDSS Mission Operations (GDSS-MO) service layer that binds them together.

Similarly service configuration and service history may be implemented using common infrastructure across multiple services. Mission database and operations archive functions can support aggregates of the corresponding data sets for all GDSS-MO services. To allow the greatest flexibility in maintenance, access, and implementation, however, the data within these aggregated functions should be structured in accordance with the service specifications.

B. Service Information Model

The service interface is based on a common information model shared by all collaborating elements in the service layer. This defines 1) the information objects that exist across the service interface, 2) the operations (shown by the symbol ⬣

in subsequent figures) that can be performed on those objects, and 3) the events (◆), or messages, that report the current status of those objects.

Each mission operations service specification has its own associated set of service object types. These object types, their attributes, operations, and associated events form part of the service specification. For more complex services (e.g., schedule execution) there may be several different types of object, with relationships between them.

For any given deployment (or instance) of the service, the actual objects that exist (parameters, commands, procedures, etc.) are specific to the mission concerned. These are defined in the configuration data for the service instance. Two basic types of service object exist: 1) statically instantiated objects, like parameters, that are fully defined in the configuration data, and 2) dynamically instantiated objects, like commands or actions, that have a definition in the configuration data, but a new instance of the object is created each time it is invoked.

Many mission operations services share the same common basis to their information model. This is illustrated in Fig. 7. For each object there are four principal elements: 1) the unique object identity; 2) static object definition information (or characteristics), such as its name, description, arguments, check conditions, etc.; 3) details for object instantiation, such as a unique instance identification (ID), time of invocation, and argument values; and 4) dynamic object status information, such as its current execution and verification status.

For statically instantiated objects, the object invocation is effectively omitted (it is a null singleton).

The uniqueness of objects and services must also be considered. In a real mission operations system, there may be many similar entities being controlled using

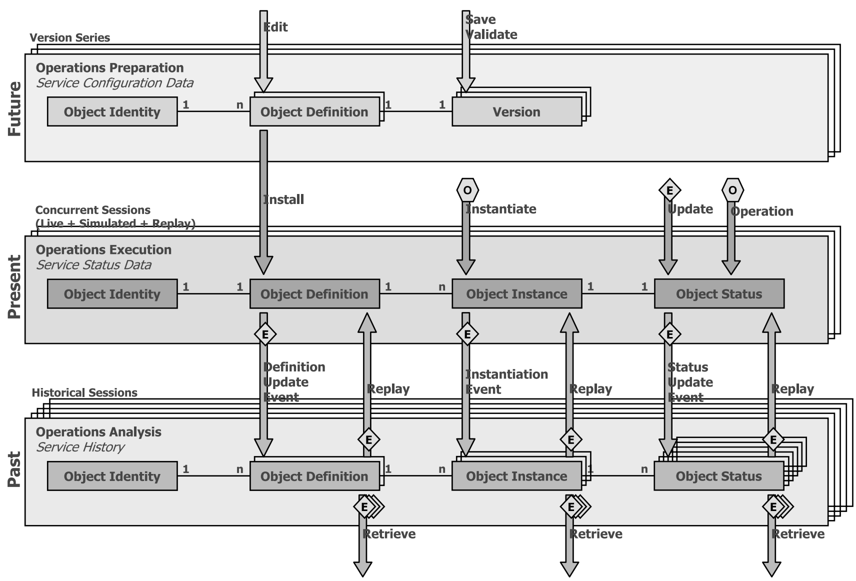


Fig. 7 Information model for a generic mission operations service object.

similar services: multiple satellites, some of similar design sharing the same parameter and command IDs. To manage this, service instances, their associated configuration data and objects, are scoped by their position within a domain hierarchy. The domain hierarchy decomposes the mission operations system into separate spheres of operational interest. Domains typically relate to real-world entities, as follows:

Agency > Mission > Spacecraft > Subsystem

Services are typically instantiated at domain level. Object identities are scoped by the domain, such that multiple objects with the same identity can exist, scoped by their domain.

Figure 8 illustrates the aspects of this information model that apply during three phases of operations.

1. Operations Preparation

It is during operations preparation that the set of objects applicable to a given mission context are defined. This corresponds to the identification of object identities and their associated definitions. This is the service configuration data for the associated service instance. These data are typically maintained using a (database) editor forming part of the operations preparation function. This is the service editor for the service. Service configuration data are maintained under configuration control, and each version will contain a set of action definitions. The service configuration data itself must conform to a standard schema that reflects the information model for the service. Service configuration data may

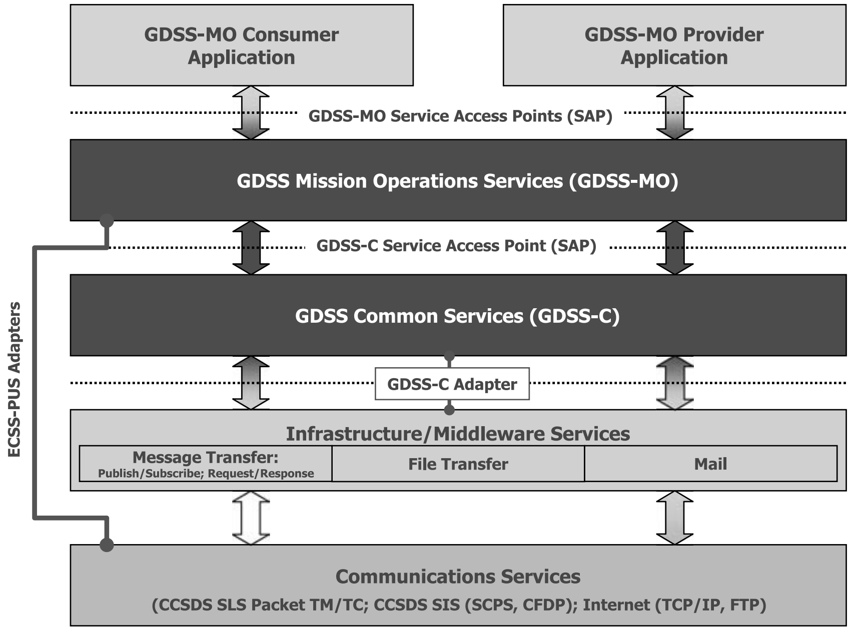


Fig. 8 GDSS service layering.

also contain references to the configuration data for other related services (e.g., a procedure definition may reference parameters or actions). To support operations execution, a version of the database is distributed and installed within the mission operations system.

2. Operations Execution

At service initialization, service objects are created for each statically defined object in the currently installed version of the service configuration data. Dynamically instantiated objects may also be created, based on the current definition, as a result of specific instantiation operations (e.g., sending a telecommand). The service also defines other operations that can be performed on objects and the events that report their status to a service consumer. Provider and consumer functions will then hold both the current definition and current status of these objects. Note that the effect of an operation is reported as an event, such that it can be observed by all consumers. Where live and simulated data exist concurrently, these are partitioned into different sessions to avoid confusion between them. Each concurrent session has its own definition and status for each parameter. A session is effectively a coherent view of the mission operations system that supports a specific operational context: a service instance may occur only once within each session, but may appear in multiple sessions with different service configuration data and status.

3. Operations History

In addition to being passed across the active service interface, the same events that report object instantiation and status to a service consumer should be available for subsequent retrieval from history. This allows the same or similar applications to work with both live and historical data. The most coherent way of ensuring that history is correctly correlated to changes in the installed service configuration data is also to store parameter definition change events in history. History is also partitioned into sessions (of which there may be many more than can be concurrently active). This leads to a logical model of history, which is structured as a tree of events.

Session > Object Identity > Object Definition > Object Instance > Object Status

History can be accessed in two main ways:

1) Retrieval. A block of events relating covering a period of time is extracted in a single transaction.

2) Replay. Discrete events are forwarded dynamically to the consumer in accordance with an evolving timeframe.

Complex information, such as operations procedures and schedules, may themselves comprise a hierarchy of different types of objects. For example, a schedule may contain predicted events, planned contacts, and planned operational tasks that are themselves broken down into a set of individually schedulable activities.

Schedule > Event|Contact|Task > Activity

Each event, contact, task, and activity has the same structure as a dynamically instantiated object—with definition, instance, and status.

If this approach is adopted systematically, although each GDSS mission operations service will have its own specific information classes, it can be seen that a common infrastructure can be devised for 1) managing the static definitions of

those objects (the service configuration data) and 2) storing and retrieving the operational history of those objects (the service history).

This data handling infrastructure could be applied to all GDSS-MO services that follow the same basic information model. If new services are defined that meet the same information model, then they can use the same infrastructure. Applications can also be developed that use multiple services with ease. A historical data replay session could also apply to multiple services and multiple consumers, allowing integrated and synchronized replay of data (parameter, command, and automation history) to provide a complete picture of what was occurring at a point in time.

C. Service Layering

A key feature of a service-oriented architecture is that it is built up in layers, starting with basic communications protocols, overlaying these with generic middleware, and ultimately providing specialized domain services. In this way, applications are isolated from the detailed implementation of the service interface, while taking benefit from existing technologies. The individual GDSS-MO services listed in Table 1 are overlaid over a GDSS Common Services (GDSS-C) layer that provides a common infrastructure supporting all or multiple GDSS-MO services.

The GDSS-C layer will provide support for the following:

1) Common mechanisms such as the service directory.

2) Common interaction patterns that isolate underlying infrastructure/middleware services, including those for message exchange, file transfer, and mail.

3) Common concepts, such as domain and session.

A benefit of implementing multiple GDSS-MO services over a smaller set of common services is that it is easier to bind these to different underlying technologies that provide the communications layers of the protocol stack. All that is required is an "adapter" layer between the common service and the underlying protocol to enable all GDSS-MO services over that technology. Hence the same GDSS-MO service can be implemented over ground-based network technologies and middleware, or even across the space link itself. The GDSS-C common layer acts as the adapter layer between the GDSS-MO services and the underlying infrastructure/middleware implementation. However, in the case of a space-ground link using ECSS-PUS, each GDSS-MO service would be directly mapped to the underlying PUS services required to implement it on the space link. The GDSS-MO services themselves provide the "plug-and-play" interface for applications, allowing them to be integrated and deployed wherever is appropriate for the mission.

D. GDSS Common Interaction Patterns

A generic structure for all GDSS-MO services has been introduced. Analysis of these services shows that a limited number of common patterns of interaction can be applied to all currently identified services. These common patterns of interaction address the active service and historical data interfaces in greater detail, and will allow more service capabilities to be provided within the GDSS-C common layer as generic services.

The following patterns have currently been identified: 1) operation, 2) operator interaction, and 3) product distribution.

The operation pattern is illustrated in Fig. 9 and discussed in more detail next. The figure shows the constituent service interfaces: active service interfaces are shown in red; service history interfaces in blue (see also the color figure section). The diamonds represent service events and the hexagons service operations. This pattern applies to the majority of identified GDSS-MO services, including: monitoring and control (MC), automation (AUT), flight dynamics service (FDS), location (LOC), onboard software management (OSM), procedure execution (PEX), planning request (PRQ), schedule execution (SEX), software reference image (SRI), and time management (TIM).

The active service interface can be divided into three components: 1) observe interface, 2) control interface and 3) manage interface.

An observe interface supports the provision of status information to any service consumer, but does not impact the basic operation of the service provider. This is achieved through a flow of status update events (◆) relating to the service interface objects. For example, a consumer of the MC parameter service will first subscribe to a set of parameters. It will then receive a flow of parameter status update events, relating to the subscribed set of parameters. It can be regarded as a "read only" interface. Many service consumers can observe the same information at the same time: a common implementation pattern is that of publish-subscribe. Note that the impact of operations performed through controller and possibly manager interfaces will be visible through the observer interface.

A control interface supports the initiation by a service consumer of operations (⬢) that are supported by the service provider. For example, a consumer of the MC parameter service may set the value of a parameter. A consumer of the MC action service can invoke an action (e.g., send a telecommand). Controller interfaces are typically one-to-one, and the number of concurrent controller interfaces may be restricted by the service provider: a common implementation pattern is that of request-response. The response may return the result of the operation or, where a new instance of an object (e.g., an action/command) has been created, the identity of the object created.

A manage interface typically concerns run-time configuration of the service provision affecting the on-going behavior of the service provider, also achieved as

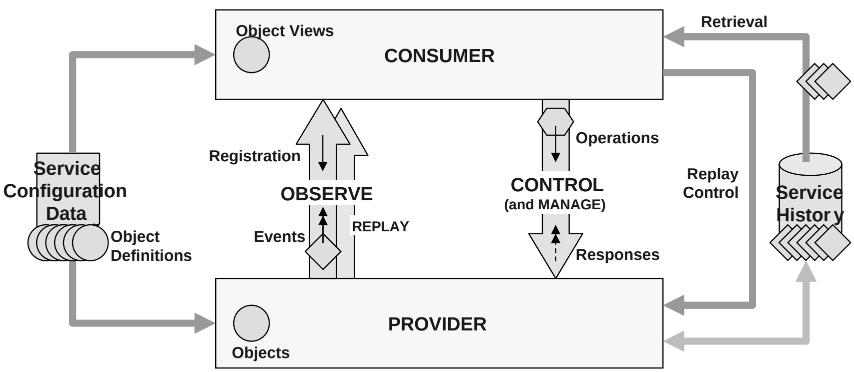


Fig. 9 Operation interaction pattern. (See also the color figure section starting on p. 645.)

operations (⬢). An example for the MC parameter service would be to disable parameter validity checking for one or all parameters. These may be regarded as specialist extensions of the control interface that are either not required or only infrequently required by most service consumers.

The service history interfaces allow access to historical data. In principle, it should be possible to access all events (◆) that could have been observed live, via the observe interface. In principle, the service history itself comprises an archive of all the observable events, logged at run-time by the service provider. However, alternative implementations that require re-processing of lower level protocol history on demand to regenerate the events are also possible. Two distinct methods of historical data retrieval are available to the service consumer:

1) Active replay: reconstruction of the observe interface for a historical time period, with dynamic replay of a sequence of discrete events (◆), in the context of a historical session.

2) Bulk retrieval: retrieval of a range of service history events (◆◆◆) (potentially meeting specified filter criteria) as a single retrieval action. A variation on this is to allow data to be retrieved in successive blocks or pages.

Active replay supports applications, such as status displays (ANDs and Mimics), that have a historical replay view. Bulk retrieval supports off-line applications, such as performance evaluation and analysis, and also status displays that show a historical trend or log view (graphical and command history displays). Active replay is supported by a replay control interface, coupled with a historical copy of the observe interface. Bulk retrieval works in a transactional way, with a request being followed by the transfer of a block of retrieved data.

IV. Conclusion

By supporting such generic interaction patterns within the GDSS-C common infrastructure layer, not only are GDSS-MO services isolated from the underlying technology, but the individual services that conform to a common pattern become a relatively thin layer. This has a number of key benefits:

1) It reduces the cost of implementing the GDSS service infrastructure, as each high-level service is little more than the configuration of the information model for the service.

2) It makes GDSS easily extensible to accommodate new or mission-specific services, as the GDSS-MO application service represents only a thin layer on top of the generic GDSS-C layer.

3) Common configuration data and archiving solutions can be developed that support all GDSS services conforming to a given pattern.

References

[1] Thompson, R. S., Cooper, S., Hall, S., and Page, O., "Definition of Standardised Ground Data System Services: GDSS Service Framework," SciSys Ltd., SSL/07947/TEN/001, Chippenham, England, UK, Oct. 2005.

[2] *CCSDS Recommended Standard for Space Data Systems*: *Mission Operations Services Concept*, Consultative Committee for Space Data Systems, CCSDS 520.0-G-2, Washington, DC, Aug. 2006.

[3] *Ground Systems and Operations – Part 1: Principles and Requirements*, European Cooperation for Space Standardization, ECSS-E-70 Part 1A, April 2000, http://www.ecss.nl.

Chapter 11

ESA Ground Operation System Architecture and the Impact on Existing ESOC Ground Data Systems

Y. Doat,[*] C. R. Haddow,[†] M. Pecchioli,[‡] and N. Peccia[§]
European Space Agency, Darmstadt, Germany

I. Introduction

A. Overview

THE European Space Operations Centre (ESOC) is responsible for the provision of ground segments for the operation and control of many ESA missions. The scope of the ground segment encompasses the elements required on ground to provide the requisite monitoring and control capabilities. This ranges from the ground stations, through the mission control and flight dynamics functionality, to the archiving and distribution of data to the science community and, in some cases, the initial processing of science data. The provision of simulators for training and operations validation purposes is also covered.

ESOC is currently carrying out an ambitious program to standardize the infrastructure used throughout its ground segments in an effort to improve the reliability, cost effectiveness, and interoperability. The end product of this program will be the ESA Ground Operation System (EGOS). Many of the systems that are used in the ground segment have been developed in the past using a variety of different technologies and policies, quite often as standalone projects that did not take into

[*]Head of the Ground Station Backend Section, European Space Operations Centre, OPS-GIB.

[†]Senior Software Engineer, Data System Infrastructure Division, European Space Operations Centre, OPS-GI.

[‡]Head of the Operation Centre System Infrastructure Section, European Space Operations Centre, OPS-GIC.

[§]Head of the Data System Infrastructure Division, European Space Operations Centre, OPS-GI.

account other developments. EGOS is therefore intended to provide a more standardized infrastructure, which encourages systematic reuse and thus reduce development and maintenance costs. The approach adopted is to use well-established software engineering principles, build on existing systems and experience, use international standards where available, and avoid proprietary systems/standards where possible.

Initially EGOS is concentrating on the areas of mission control systems, simulators, and some of the ground station equipment, in particular the base-band systems and ground station monitoring and control. In the longer term it is envisaged that the scope will be extended to cover satellite electrical ground support equipment (EGSE), flight dynamics and mission planning systems. EGOS must also permit easy expandability, as it is not possible to forecast with certainty future system requirements.

The approach adopted in the architecture is one of a service-oriented system based on component application development. The service and component architectures represent two views of the system and are therefore not mutually exclusive, but complementary. Within the overall architecture categories of components have been identified. The first of these are the core components; these provide the essential services and functionality on which the other components are built. Next come the common components; these provide widely used services such as file archiving, user management, etc. The core and common components can effectively be thought of as middleware for the ground data systems.

The third category, custom components, consists of the applications that implement the end user needs. These map roughly onto subsystems, e.g., telemetry chain, telecommanding chain, onboard software maintenance system, etc. Encapsulating the overall EGOS architecture is the service management framework (SMF). This provides the mechanism through which the underlying systems expose services to external users and systems in a well-defined and controlled manner. To avoid unnecessary dependencies, a set of rules has been prepared specifying whether the different categories of components may utilize the services provided by other components categories.

B. Rationale

The currently existing infrastructure software works well, as can be seen from its use in a considerable number of successful missions, recent examples being Rosetta, Mars Express, Smart-1, and Venus Express. In view of this, one might be tempted to ask, “If it’s not broken, why fix it?” The following points summarize the answer:

1) Heterogeneous operating systems (Unix, Linux, NT, Windows 2000, QNX), i.e., lack of common approach on operating system policy.

2) Heterogeneous hardware platforms (Alphas, SUNs, PCs, VMEs), i.e., lack of common approach on hardware vendor independence.

3) Lack of standard human computer interface (HCI) look and feel.

4) Lack of standard network and communication services.

5) Lack of standard language or protocol for data interchange.

6) Lack of common metadata model.

7) Lack of common standard for event and log messaging.
8) Lack of common standard for data access (files and databases).
9) Lack of standard of databases used across subsystems.
10) Lack of standard internal representations for data.
11) Lack of common approach to system monitoring and control.
12) Lack of common approach to security.
13) Lack of common approach to fault management.
14) Lack of common approach to configuration and system initialization.
15) Lack of unique software maintainability approach.
16) Lack of isolation of software systems from operating systems and COTS to improve portability.
17) Lack of clear isolation between components of a subsystem.
18) Lack of synergy across developments.
19) Proliferation of test tools to support the validation of the various subsystems.

From the preceding there is obviously a need to rationalize the software development across the different ground segment components. With this in mind EGOS is the architectural framework for ground segment systems that aims to provide this rationalization and has the following goals:

1) Definition of a reference architecture for the ESA ground operations software.
2) Establishment of a core infrastructure for ESA ground segment systems.
3) Standardization of the interfaces between components.

II. Scope

The scope of the EGOS architecture is intended to encompass, in the long term, almost all of the scope of an ESA ground segment system. Initially the main focus of activity is concentrated on the ground stations, control systems, simulators, and the automation of operations. Once the initial architecture has been proved in these domains, it can then be expanded to cover additional systems such as flight dynamics, mission planning, electrical ground support equipment (EGSE), etc. The following sections outline the main areas that will be covered by EGOS.

A. Initial Scope

An ESA ground station is composed of a front end that drives the antenna and deals with the signals and of a back end that ensures the digital data computation and communication with the control center. Various devices and computers, including telemetry and telecommand processors [e.g., telemetry and telecommand system (TMTCS)], frequency converters, and switches, ensure the uplink and downlink communications between the ground and the spacecraft, the telemetry (TM) and telecommand (TC) production.

All of these devices are monitored and controlled to ensure the proper configuration and operation of the ground stations. The station computer (STC2) centralizes all of the information and enables the remote monitoring and control (MC)

from the control center. The monitoring and control functions cover reading and writing of variables, handling of tasks, retrieval of logs, and setting of administrative states. The STC2 interfaces directly with some of the subsystems and relies on two modules, the front-end controller (FEC) and the monitoring and control module (MCM) for the MC of specific front-end devices and back-end devices, respectively. With this in mind, it is intended that the following elements are incorporated:

1) Base-band system at the ground station, i.e., the systems to which the operations control center (OCC) interfaces to receive telemetry and transmit telecommands.

2) Ground station monitoring and control.

With respect to the control systems, it is intended that EGOS encompass all aspects of what is covered by an ESOC mission control system (MCS), namely:

1) Control of the interface between the OCC and the ground station using the Consultative Committee for Space Data Systems (CCSDS) Space Link Extension (SLE) protocols thus enabling interoperability with non-ESA ground stations that support this.

2) Monitoring and control of the spacecraft.

3) Monitoring and control of the ground segment.

4) Archiving and distribution of data.

5) Provision of services to external users.

As noted before, part of the responsibility of an ESA ground segment is to provide simulators capable of supporting procedure validation activities and training needs. In addition, certain simulation functions are required at the ground stations to enable dataflow tests between the OCC and the ground station to be carried out. EGOS is thus required to provide the following facilities: 1) infrastructure software supporting the development of spacecraft simulators and test tools, and 2) portable satellite simulator (PSS) for use during all phases of ground systems validation.

The area of mission automation must also be addressed. There is currently ongoing work at ESOC on the design and implementation of an automation system that will enable some automation of mission and ground station operations. The definition of this automation system is being carried out within the scope of the overall EGOS architecture and is described in more detail in [1].

B. Longer Term Scope

The EGSEs for a number of ESA spacecraft are already based on a modified version of the existing mission control system software. Because of the specialized nature of the equipment that is used in testing the spacecraft, the main modifications required to use MCS software are related to the interfacing to this equipment. EGOS should therefore aim to provide a flexible and extensible interface that will simplify the connectivity to the spacecraft test equipment. In addition, it is likely that, because of the hard real-time requirements of some safety critical aspects of EGSE operations, there may be some special components required.

Currently mission planning systems (MPS) at ESOC are largely custom implementations for each mission, possibly reusing some of the code base from existing

missions, but modifying this to fit the needs of the particular mission. The goal of EGOS with respect to MPSs would be to provide a set of specialized components in addition to the common and core components that could be used as a toolkit to implement a planning system.

Flight dynamics systems are currently, because of their specialized nature, not covered by EGOS. The long-term goal would be to integrate these more fully, initially using the external interfaces provided by the service management framework to prove access to control system functionality. Later the intention would be to migrate the specialized applications to use the common and core functionality provided by EGOS.

A key area that must also be addressed by the system is expandability. As mission demands are constantly evolving, it is essential that the architecture be readily expandable to cover new requirements and to permit the inclusion of additional functionality.

III. EGOS Architecture

The EGOS architecture incorporates a basic middleware infrastructure, reusable components, development guidelines, and tools that together provide a consistent approach to the development and deployment of mission ground systems. An overview of the EGOS architecture is shown in Fig. 1.

It can be seen from Fig. 1 that the EGOS architecture has four main elements: the EGOS framework, EGOS components, EGOS user desktop, and service management framework (SMF). These elements are described in more detail in the following sections. The figure also shows non-EGOS applications, which are internal elements of the ground system and which interact with EGOS applications, but do not themselves run within the EGOS run-time environment. They are discussed in more detail in the following sections. External applications are external to the mission ground system and are consumers of EGOS ground system services. These are also discussed in more detail in the following sections. It should be noted that EGOS applications may interact with other EGOS applications via the SMF, i.e., they could also play the role of consumers of EGOS ground systems services.

A. EGOS Framework

The EGOS framework comprises a number of elements that support the development, deployment, configuration, and execution of EGOS applications. In the following section the concept of the EGOS component, from which EGOS applications are built, is introduced. The main elements of the framework, the EGOS component run-time framework, the EGOS deployment and configuration framework, the EGOS development framework, and the EGOS class libraries, are then introduced.

1. EGOS Components

EGOS applications are built from components and typically provide some high-level function within the mission ground system. There are many definitions

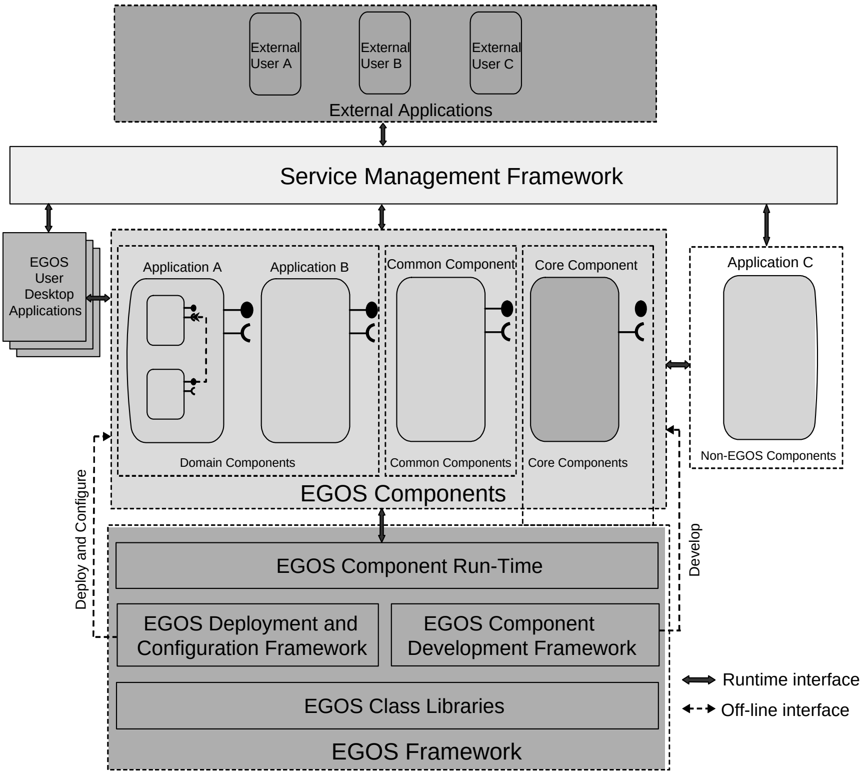


Fig. 1 EGOS architecture overview.

of a component, but for the purposes of an EGOS application, the following definition shall be used: A component represents a modular part of a system that encapsulates its contents and whose manifestation is replaceable within its environment. A component defines its behavior in terms of provided and required interfaces. Larger pieces of a system's functionality may be assembled by reusing components as parts in an encompassing component or assembly of components, and wiring together their required and provided ports.

EGOS applications are component-based applications. In terms of the EGOS framework, an application is just a component that may itself be composed of components and provides some independently useful function within the ground system. Figure 2 shows the two main types of EGOS components and their relationship to the EGOS development, deployment and configuration, and run-time frameworks.

The two types of components identified in Fig. 2 are as follows:

1) Monolithic component—components that are compiled code (called monolithic implementations).

2) Component assembly—assemblies of other components (assembly implementations, providing a recursive definition).

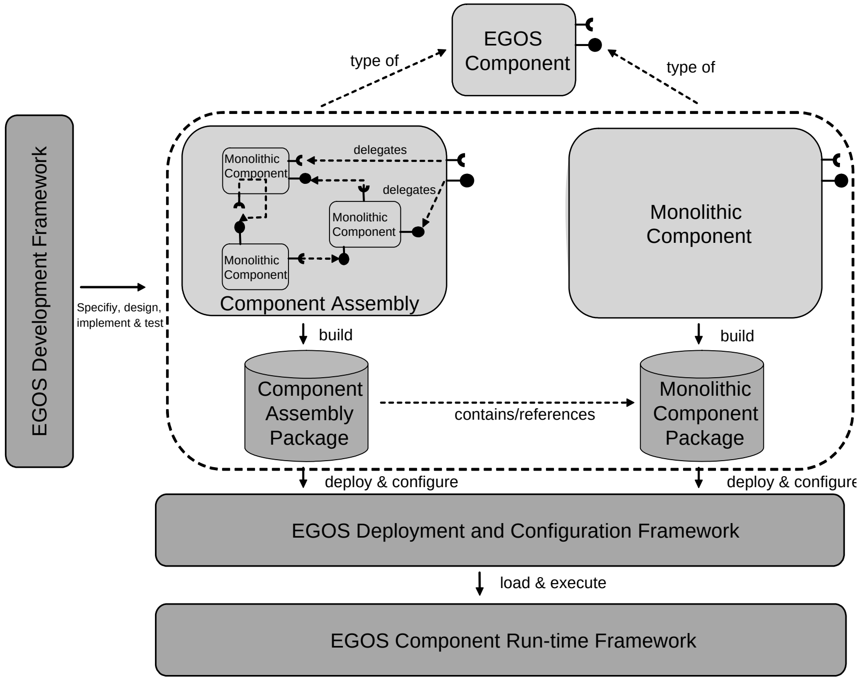


Fig. 2 EGOS component overview.

A component assembly is defined as a set of components and interconnections that implement a component. A component assembly is itself a component, with specified required and provided interfaces that are delegated to its contained components. There is no special "top-level assembly," since assemblies are simply a method of specifying component implementations. To actually execute a component whose implementation is an assembly of lower-level components, there must eventually be monolithic implementations at the "leaves" of the hierarchical implementation. An EGOS application may be either a component assembly or monolithic component.

EGOS will define a standard packaging approach for components, so that components can be easily deployed, configured, and launched within an EGOS ground system. A component package is the minimal unit of deployment and consists of a set of metadata and compiled code modules that contain implementations of a component interface. The implementations in a package can be a mix of monolithic and component assembly implementations, with either or both present at any level of the hierarchy. The creator of a component-based EGOS application produces a component package whose top-level component interface represents the interface of the application. A monolithic component can run on only a single node, while a component assembly may run on either a single node or can be distributed across multiple nodes.

2. EGOS Component Run-Time Framework

An EGOS component runs within a component server, based on the EGOS component container framework, as shown in Fig. 3.

The EGOS container framework defines the application programming interface (API) of the run-time environment in which a component executes. It therefore enables components to run on any platform that provides an implementation of the container framework. The container framework also allows extensions (e.g., hard real-time support could be provided by real-time extensions to the container).

The container is responsible for such concerns as component life-cycle management and the execution of component assemblies within a distributed environment. Figure. 4 shows an example of a component assembly that has been deployed onto multiple nodes and processes within the target environment.

Management for launching, execution, and termination of EGOS applications is provided through an application manager. The application manager takes as input the "deployment plan" for the application and uses the information within it to instantiate and configure the application's components within the target environment. When the deployment plan requires that the component assembly is distributed on different network nodes, the application manager delegates the management of component instantiation, execution, and termination to node application managers.

3. EGOS Deployment and Configuration Framework

The deployment and configuration of EGOS applications within a distributed environment is a critical activity that is supported by the EGOS deployment and configuration framework. It is assumed for the deployment and configuration process that a component package has already been delivered by a development team, developed with the support of the EGOS development framework. There is a target environment, consisting of a distributed system infrastructure (computers, networks, services), on which the software will ultimately run, and there is a repository, which as a minimum, is a staging area where the packaged software

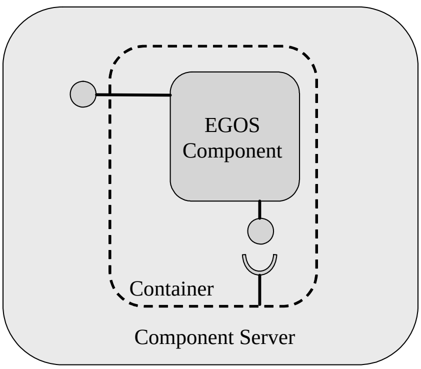


Fig. 3 EGOS component server.

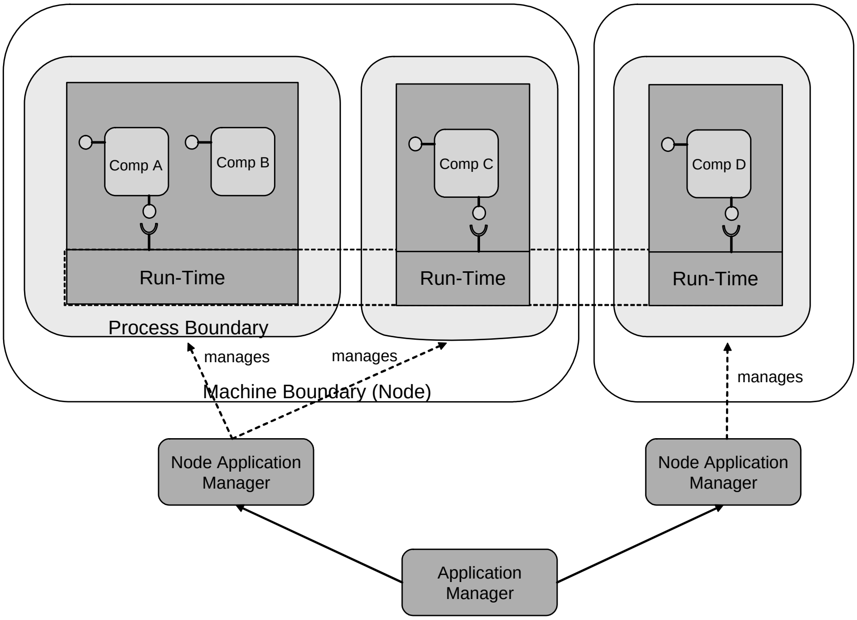


Fig. 4 Component distribution infrastructure.

is captured prior to decisions about how it will run in the target environment. Figure. 5 shows an overview of the deployment and configuration process.

There are a number of logical steps in the deployment and configuration of an EGOS application in the target environment:

1) Installation and configuration. Installation is the act of taking the software package coming out of the development process and bringing it into a component software repository under the deployer's control, but where the location (computer, file system, database) of this repository is not necessarily related to where

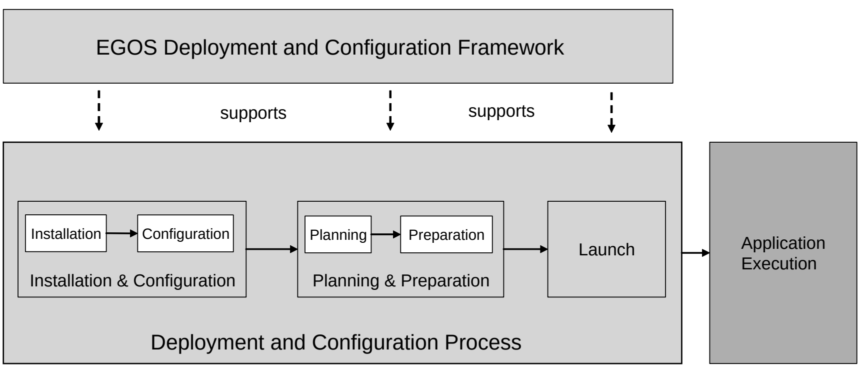


Fig. 5 EGOS deployment and configuration framework.

the software will actually execute. When the software is in a repository, it can be functionally configured as to various default configuration options for later execution.

2) Planning and preparation. Planning how and where the software will run in the target environment is an activity that takes the requirements of the software to be deployed, along with the resources of the target environment on which the software will be executed, and deciding which implementation and how and where the software will be run in that environment. Planning is deciding how and where the software will run. Preparation is performing work in the target environment to be ready to execute the software, such as moving binary files to the specific computers in the target environment on which the software will execute.

3) Launch. Launching the application brings the application to an executing state. Component-based applications are launched by instantiating components as planned, on nodes in the target environment. Launching includes interconnecting and configuring component instances, as well as starting execution. In this executing state, the application runs until it completes or is terminated.

To support the preceding steps, the following are required for the deployment of software components into a distributed environment:

1) Component metadata (manifest). Describes the packaged component assembly and its interconnections (i.e., the application). The component metadata (or manifest) is part of the deployed software package.

2) Target data model. Describes the target environment in which the assembly of components contained in a software package are to be executed (i.e., nodes, inter-connectors, etc.).

3) Deployment plan. Describes how the components of the application are to be allocated to nodes in the target environment. Different deployment plans may therefore be produced for the same application and target environment (see Fig. 6).

An EGOS application manager is used to launch and later terminate an EGOS application according to the deployment plan. When the EGOS application is distributed, it must manage the pieces of the application (i.e., components) that run on each node (i.e., node application manager).

4. EGOS Class Libraries

The EGOS class libraries provide validated implementations of common functions and algorithms used across a number of EGOS components, thereby reducing the effort and increasing the reliability of component implementations. The use of a class library is an internal detail of a component implementation and is not visible to the component's environment (i.e., the EGOS framework and other EGOS components).

B. Component and Service Categories

The EGOS architecture identifies a number of standard EGOS components and services. Figure. 7 presents how EGOS components are characterized.

EGOS services define the required and provided interfaces of EGOS components. Services and components are characterized as follows:

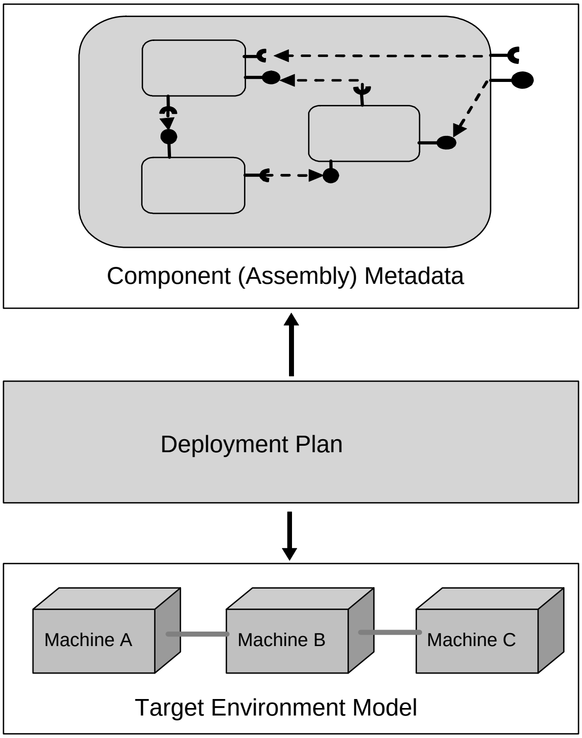


Fig. 6 Deployment plan.

1) Core. Core services are those guaranteed to be provided by the EGOS framework. Core services can be implemented by core components, although this is an internal implementation detail of the framework. The core services are 1) service directory, 2) configuration access, 3) event distribution, 4) action execution, 5) session management, and 6) security.

2) Common. Common services are those that are implemented by common components and whose function is not specific to a particular domain (see the following for description of domains). Common services are not guaranteed to be provided by the EGOS framework and therefore a deployment of an application that depends on common services must ensure that the common components that implement these services are also deployed in the target environment. A common component shall have no dependencies on custom services or components.

3) Custom. Custom services and components are specific to a particular application domain. A domain is characterized by its high-level function within the ground system (i.e., monitoring and control, simulation, flight dynamics, ground station functions, etc.). Custom applications can use services from other domains, but may not be composed of component implementations from other domains. A custom application may be composed from either common components or components from its own domain.

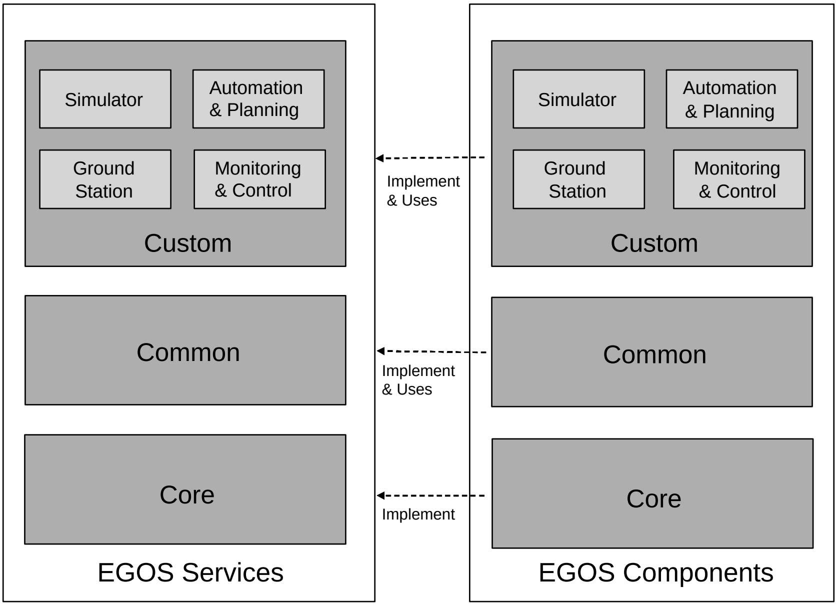


Fig. 7 Component and service categories.

In addition to the characterization of EGOS components just presented, it is also possible to assign components to the different tiers of an n-tier architecture.

C. Service Management Framework

The SMF provides an abstraction layer that allows clients to transparently monitor and control systems elements. Clients may be both external systems (e.g., an external user such as a principle investigator who wishes to access the functionality of the system from a remote location, for example to retrieve telemetry data from their instrument) or internal EGOS applications (e.g., the mission automation system). The system elements being monitored and controlled may be internal EGOS applications or external systems (e.g., a spacecraft undergoing assembly, integration, and verification activities). The SMF effectively provides a monitoring and control service bus, where clients can transparently access the monitoring and control services of both space and ground system elements. Figure. 8 shows the relationship between the SMF and EGOS applications and clients.

The SMF monitoring and control concept is based on the ECSS-E-70-31 specification, which introduces the concept of a space system model (SSM) as a means for capturing mission knowledge used during assembly, integration, and verification (AIV) and operations. This knowledge is used by the different monitoring and control applications to interact with the space system and to process the

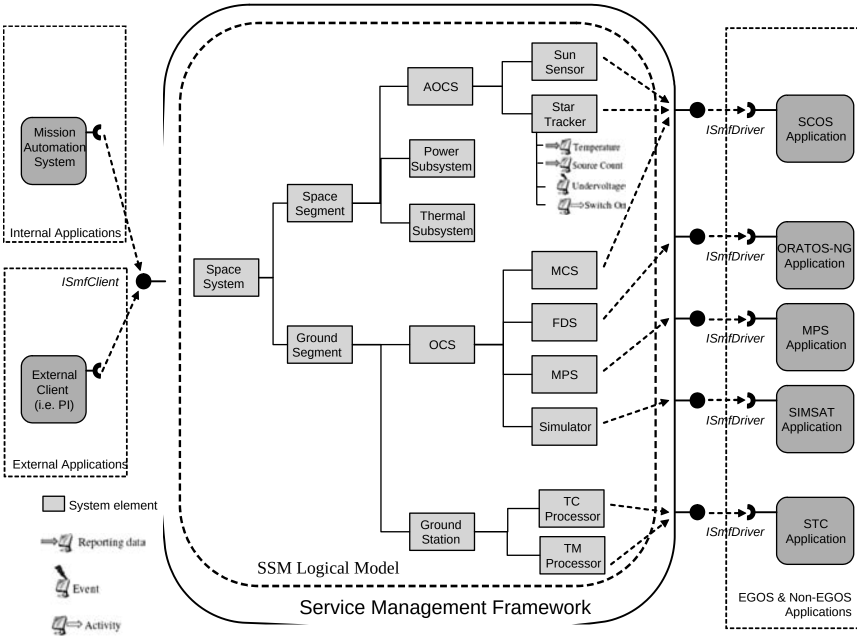


Fig. 8 Service management framework.

dynamic data that are exchanged with it (i.e., space segment telemetry and telecommands, ranging data, ground segment commands, and measurements).

The SSM consists of different types of objects and the relationships between these objects. The objects of relevance are system elements, reporting data, activities, and events:

1) System element. This includes any system element (both ground and space segment) that can be monitored and controlled. A system element can have reporting data, activities, and events associated with it (see the following).

2) Reporting data. Reporting data comprises parameter values that have a meaning for the monitoring of the space system (e.g., operational mode of a ground station application, temperature measured by a thermistor onboard the spacecraft, etc.).

3) Activity. An activity is a space system monitoring and control function. Examples of activities could be a telecommand (either to the space segment or to the ground segment) or ground system task (e.g., a printer request, sending an e-mail, etc.). An activity can be performed over a period of time, be comprised of a number of actions, and have state.

4) Events. An event is an occurrence of a condition or set of conditions that can arise during the course of a test session or a mission phase (e.g., loss of a spacecraft telemetry signal by the ground station, under voltage of a unit onboard the spacecraft, etc.). It can be used to trigger monitoring and control actions implemented within the space system.

Clients of the SMF are able to transparently access system elements and their associated mission knowledge (i.e., reporting data, activities, and events). An SMF client is therefore able to monitor reporting data, initiate and monitor activities, and receive notification of events of any system element in the space system element.

The SMF interacts with EGOS applications though SMF application drivers that perform the monitoring and control services requested by SMF clients. The SMF delegates responsibility to the applications that implement the SMF application driver interface via the following functions on a given set of system elements: 1) get reporting data values (real-time and historic), 2) initiate and monitor activities (real-time and historic), and 3) receive notification of events (real-time and historic).

The binding between a system element and an SMF application driver interface is done dynamically and defines how the logical view of the system under control (i.e., space system model) is mapped to the physical system (i.e., ground system software applications). For example, the spacecraft system elements (and subelements) and their associated reporting data, activities and events would generally be mapped to SMF application driver of SCOS-2000, while the ground station elements (and subelements) and their associated reporting data, activities, and events would be mapped to the SMF application driver of the STC.

Deployment of the SMF is foreseen to be on a mission-specific basis, with the scope eventually extending to cover all of the mission-specific systems. In the future it is envisaged that each ground station may have its own particular SMF that controls access to the services provided by the ground station.

It should be noted that there is some overlap between the SMF concept and the current work being carried out by CCSDS on spacecraft monitoring and control standards.

D. EGOS User Desktop

The EGOS user desktop provides a framework for EGOS Graphical User Interface (GUI) applications (presentation tier) and allows the monitoring and control of applications that run within the EGOS framework. The user desktop allows the addition of new functionality by means of plug-in components that execute within the user desktop run-time. It is envisaged that a number of standard plug-ins will be provided (data viewers, log viewers, user login controls, etc.) that are shared across desktop applications and that reduce the overall desktop application development and maintenance effort while allowing greater flexibility in the configuration of the desktop to satisfy different user requirements. It shall also provide a consistent look and feel for all EGOS desktop applications. Figure. 9 shows the high-level architecture of the EGOS user desktop.

EGOS user desktop applications are able to access EGOS applications through either the SMF or directly with an EGOS application by accessing the application's service(s) via the EGOS service directory. It is envisaged that the EGOS user desktop framework will be based on a standard application framework and that EGOS will augment the standard application framework with standard EGOS plug-ins and an EGOS application style guide. The desktop plug-ins will consist of both graphical plug-ins [e.g., command stack plug-in, alphanumeric display

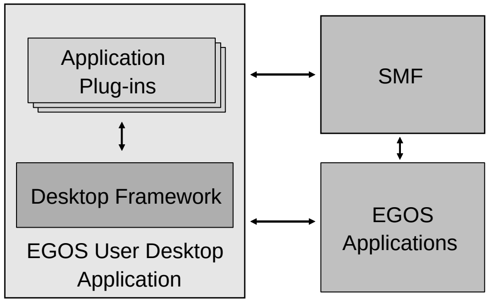


Fig. 9 EGOS user desktop applications.

(AND) plug-in, etc.] and non-graphical plug-ins (e.g., plug-ins to access standard EGOS application services or the SMF, etc.).

E. Non-EGOS Applications

A non-EGOS application is one that does not run within the EGOS component run-time. The development, deployment, configuration, and execution are an internal detail of the application that is outside of the scope of the EGOS. A non-EGOS application may, however, be an integral part of an EGOS-based ground system, and therefore interoperability of the non-EGOS application with other EGOS applications or the SMF is a critical issue. Figure. 10 shows the basic approach for interoperability between EGOS and non-EGOS applications, which is that a non-EGOS application interacts with EGOS applications or the SMF through an EGOS service adapter layer. The adapter layer translates interactions between standard EGOS services and the SMF and the non-EGOS application. The adapter layer may be implemented as an extension to a non-EGOS application or may be performed by a broker component that sits between an EGOS application

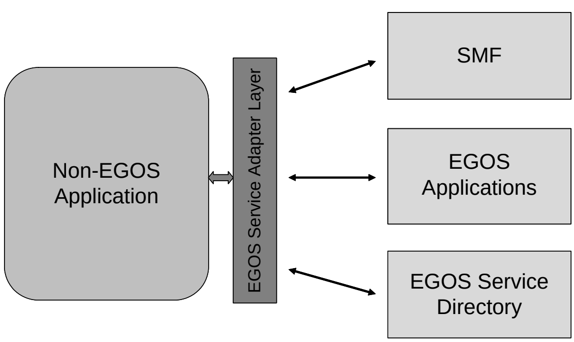


Fig. 10 Non-EGOS application.

and non-EGOS applications, where no modifications to either application is necessary.

As the re-engineering of existing ground system applications or products to EGOS is expected to be an evolutionary process, it is envisaged that initial ground system deployments will consist of a mixture of EGOS and non-EGOS applications. Initial deployments will therefore require the development of EGOS service adapter layers that will later be replaced when the involved non-EGOS are eventually re-engineered.

F. External Applications

External applications are external to the EGOS ground system (e.g., an application of a principle investigator that wishes to access the functionality of the system from a remote location, for example to retrieve telemetry data from an instrument). An external application is able to use services of the EGOS ground system but does not perform any functionality required for the ground system to perform its function. External applications may also interact with an EGOS ground system across insecure public networks (see Fig. 11).

It is envisaged that interactions between external and internal EGOS applications will be performed through the service management framework interface—the SMF will make available services to which the external application has been granted access. The SMF will therefore authenticate all service requests and ensure that sensitive data passed across the interface are secure.

IV. Evolution of MCS and Simulators Infrastructure Toward EGOS

A strategy for the evolution of the existing control center infrastructure software toward an EGOS-based architecture has been defined, the main principles of which are briefly summarized as follows:

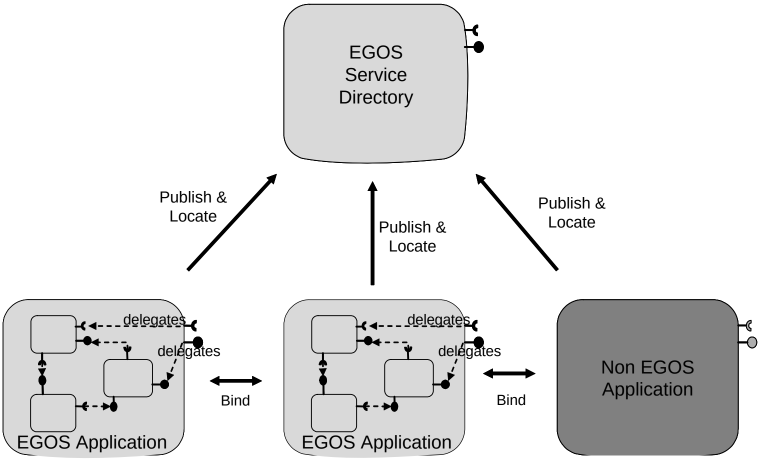


Fig. 11 EGOS applications in a service-orientated architecture.

1) Only elements that are outdated in terms of technology used and/or supported functionality will be redeveloped from scratch.

2) The architecture of existing elements that have to be "migrated" to an EGOS-based approach (i.e., not rewritten) will be progressively modified to prepare for the adoption of EGOS components as soon as they become available.

The preceding approach aims at minimizing the effort involved for the development of EGOS-based control center infrastructure as well as maximizing the reuse of EGOS components. The following sections provide specific details of the envisaged evolution for MCS infrastructure.

A. MCS Infrastructure

The existing mission control system infrastructure at ESOC consists of the well-known SCOS-2000 system and a variety of other ancillary systems interfacing with it. The technology being currently used is pretty much in line with the one adopted for EGOS components, except in the area of graphical user interfaces (which is currently implemented in C++ and relies at run-time on the availability of a commercial tool). Looking at the high-level architecture, it is noted that the strictly layered approach imposed by the EGOS architecture has not been systematically adopted in all areas. In particular, client applications are not properly "split" into a business layer (implementing the processing logic) and a visualization layer (implementing the user front-end interface). Going into a deeper level of detail in the analysis of the existing architecture, it becomes obvious that many of the existing low-level components support functionality that largely overlaps with the corresponding components of the EGOS architecture. These components will eventually have to be replaced by the equivalent EGOS-based ones. However, it is not excluded that for some of these components the existing implementation in the MCS infrastructure will actually be used as the starting point for developing the EGOS component, thus ensuring maximum reuse of "working" elements.

Based on the main considerations just summarized, the following MCS infrastructure evolution steps have been defined:

1) Introduce strict layering in the existing implementation. This step will lead to the "partitioning" of SCOS-2000 into these four main elements: 1) MCS framework, i.e., the set of core services required by all applications involved in space systems monitoring and control; 2) a set of processes performing the processing required by mission control systems (typically involving handling of TM/TC data); 3) a set of GUIs supporting the user interface with the underlying components, in particular with the TM/TC data processors; and 4) a set of drivers enabling the usage of the EGOS service management framework to expose services to other ground data systems (e.g., for automation or data dissemination).

2) Develop a new generation of monitoring and control GUIs. This step aims at a complete replacement of the existing legacy implementation of graphical user interfaces with Java-based, more modern implementations that will act as plug-ins of the EGOS user desktop and will make use of the same look and feel and common widgets as any other EGOS-based ground data system.

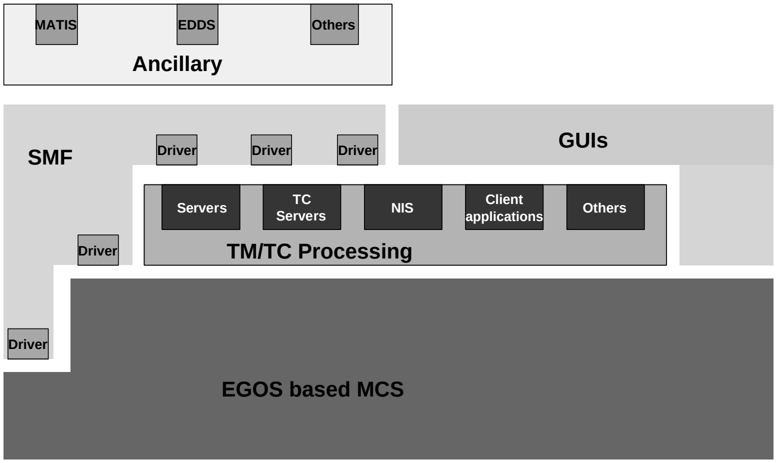


Fig. 12 MCS infrastructure-based architecture.

3) Develop a new generation of ancillary systems, supporting the data dissemination (EGOS data dissemination system) and the automation of mission operations (mission automation system). These new sets of ancillary systems will be implemented on the basis on any available EGOS low-level components and will interface with SCOS-2000 in a way that will ensure complete decoupling from the underlying implementation.

4) Migrate the MCS framework to EGOS-based implementation. This step involves the replacement of existing low-level components with the corresponding ones delivered as part of the EGOS framework. It is envisaged to “wrap” the EGOS APIs to minimize the impact of this migration onto other components using the affected low-level services.

Fig. 12 provides a high-level overview of the mission control system infrastructure EGOS-based architecture.

B. Simulator Infrastructure

Similar considerations to the preceding ones can be made to define the evolution of the existing simulator infrastructure. However, the evolution of the simulator infrastructure toward EGOS will imply less severe changes to the legacy elements, mainly for the following reasons:

1) The technologies used by the latest generation of ESOC simulator infrastructure are completely in line with the ones promoted/adopted by EGOS.

2) The architecture of the simulator infrastructure is already component based and strictly layered.

3) The simulator infrastructure, for its very nature, relies on a significant number of elements that are specific to this domain and thus not supported by EGOS components (e.g., support of adapters for models complying with the simulation model portability standard, library of models simulating space systems elements).

Nonetheless, the adoption of EGOS will lead to a non-negligible impact in the following areas:

1) Use of EGOS components providing services required by the simulators infrastructure. So far, the following EGOS services have been identified as potentially interesting: 1) events logging (archiving, distribution, retrieval, and visualization); 2) users authentication (currently not supported in the simulators infrastructure); 3) files management; 4) configuration access; 5) service directory; and 6) GUI plug-ins framework and generic plug-ins (EGOS user desktop).

2) Use of the EGOS standard libraries. This will affect the existing generic models, primarily in the area of TM/TC data management.

V. Evolution of Ground Station Infrastructure Toward EGOS

The evolution of the currently deployed ground station system must fulfill the following criteria: 1) evolution of the current systems and not revolution of the entire approach; 2) migration of the existing system tailoring to the new system; and 3) deployment of the new system without disturbing on-going mission support.

The evolution of the ground station infrastructure toward EGOS will result from a careful analysis of the existing architecture. This will enable a plan for the migration from the existing architecture into an EGOS-compliant architecture to be devised. EGOS components will be compared with the identified existing functions so that they can be integrated into the system while maintaining its essential functionality. The service approach of the EGOS architecture may also enlarge the current capability of the system by: 1) adding new functions such as data archiving; 2) providing, possibly, a different approach to servicing remote MC clients by using the EGOS service management framework; 3) implementing a harmonized look and feel based on the EGOS user desktop; and 4) integrating the EGOS log function.

VI. Conclusion

The status of the various facets of EGOS is summarized as follows:

1) Revision of EGOS high-level architecture is ongoing. This is concentrating on expanding the initial draft to provide better coverage of the simulators and ground stations.

2) The service management framework is provisionally accepted.

3) New systems (e.g., network interface system [2], ESTRACK management system [1]) are being developed based on existing components.

4) Partitioning of SCOS-2000 is ongoing as part of the Release 5 development.

5) EGOS LLC and framework (core components) software requirements have been reviewed and finalized.

6) Architectural design for core components is expected to start in 2006.

7) User desktop software requirements are currently being finalized. Prototyping activities of certain aspects of the architecture are ongoing.

8) Identification of candidate common components from the various domains has started.

The EGOS architecture will provide the development framework for the next generation of ESA ground segments. Because of the investment in existing systems, an evolutionary approach has been adopted that will migrate the majority of systems in ESA ground segments onto a common infrastructure. The overall architecture is currently being finalized and appropriate technologies are being chosen; however, it is already sufficiently clear to allow new developments to be started that will readily integrate into it.

Indeed, development of the first EGOS subsystems are already well advanced. For example, the network interface subsystem [2], which will be responsible for control of the interface between the OCC and the ground stations based on SLE services, is currently undergoing initial testing. This has been implemented as a component in such a way that it can be integrated completely into a SCOS-2000-based control system or run as a standalone system utilizing a minimum subset of SCOS-2000 that is capable of supporting legacy missions or external users.

References

[1] Calzolari, G. P., Doat, Y., Haag, S., Haddow, C. R., Pecchioli, M., and Sørensen, E. M., "Automation of ESOC Mission Operations," *SpaceOps 2006 Conference*, Rome, June 2006.

[2] Haddow, C. R., Müller, C., Scotti, G., and Ulriksen, T., "Network Interface System (NIS); ESAs Next Generation CCSDS SLE Interface System," *SpaceOps 2006 Conference*, Rome, June 2006.

Chapter 12

ESA Deep Space Antenna 2 Pointing Calibration System

R. Osterried*
SED Systems, Saskatoon, Saskatchewan, Canada
P. Droll†
European Space Agency, Darmstadt, Germany
and
R. A. Plemel‡
SED Systems, Saskatoon, Saskatchewan, Canada

I. Introduction

IN 2005, Systems Engineering Division (SED Systems) of Calian Ltd, Saskatoon, Canada, installed and commissioned a pointing calibration system (PCS) for the second ESA 35-m antenna with X/Ka-band telemetry, tracking, and command (TTC) at Cebreros, Spain. The large antenna diameter together with the operation at Ka-band results in challenging pointing requirements. The system fully automates pointing error measurements and provides an easy to use tool for the calculation of systematic pointing error model coefficients. The PCS takes into account systematic pointing error sources such as gravity effects, misalignments, non-orthogonalities between axes, beam waveguide mirror alignment errors, radio frequency (RF) beam squint, etc. Also, the PCS calculates and applies a correction for the thermal distortion of the main reflector and subreflector struts.

The PCS is fully automated and is under remote control from the European Space Operations Centre (ESOC) in Darmstadt, Germany. The pointing error requirement of the antenna is 5.5 mdeg (3-sigma including measurement error) under worst-case environmental conditions (45 km/h average wind speed, gusting

*Systems Engineer, Engineering Division, SED Systems.

†Antenna and Servo Systems Engineer, Ground Station Engineering Division, European Space Operations Centre.

‡Lead Systems Engineer DSA2 Project, Engineering Division, SED Systems.

to 60 km/h, and over a −20°–50°C temperature range). This requirement is based on the maximum permissible gain loss to support X- and Ka-missions. At Ka-band, the loss due to 5.5 mdeg pointing error (PE) is 1.2 dB. X-band pointing losses are typically less than 0.1 dB. Figure 1 is a plot of Ka-band pointing loss vs pointing error.

The PCS is fully integrated with the servo subsystem that applies the pointing corrections. The system is designed to measure the antenna's systematic pointing errors, and together with other parts of the servo system, compensate for them. Corrections are polarization (pol) dependent for either left-hand circular polarization (LHCP) or right-hand circular polarization (RHCP). Time-independent systematic pointing error sources include 1) gravity deformation of the main and subreflector as elevation angle changes, 2) azimuth (Az) encoder offsets, 3) azimuth/elevation (Az/El) axis non-orthogonality, 4) antenna tower tilt, 5) RF collimation, 6) beam waveguide mirror and feed alignment errors, and 7) RF beam squint (polarization and frequency band dependent).

Pointing errors that are actively compensated include 1) thermal gradients in the reflector back-structure and subreflector struts as measured by the temperature measurement system (TMS); 2) atmospheric refraction according to current ambient temperature, pressure, and humidity and a refraction model; and 3) tilt of the tower and azimuth structure because thermal loads are measured using tiltmeters at the elevation axis level of the antenna.

The impact of mechanical deformation due to quasi-constant wind has been limited by a mechanical design optimized with respect to maximum stiffness. Realistically, no active compensation is possible. The impact of wind gusts is reduced by a rigid and powerful antenna drive and a state-of-the-art servo controller. This and other dynamic, random pointing errors or compensation errors result in the true pointing error of the antenna. Table 1 indicates the approximate magnitude of the main pointing error sources before correction.

The pointing calibration system uses a radiometer, which can be operated as a total power radiometer (TPR) or as a noise adding radiometer (NAR) at X- and

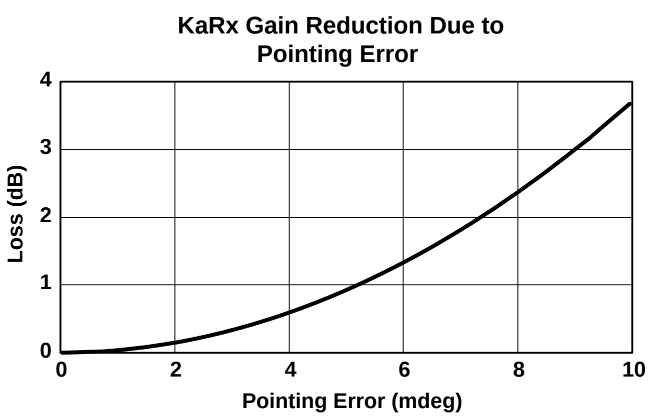


Fig. 1 Ka-band (32 GHz) pointing loss.

Table 1 Pointing error magnitudes

Pointing error source	Approximate magnitude	Estimation method
Refraction	85 mdeg at 10 deg elevation 300 mdeg at 2 deg elevation	Modeled, measured
Gravity	80 mdeg	Measured
Misalignments, non-orthogonalities, RF beam squint	15 mdeg (total)	Measured
Thermal deformations of azimuth housing	6 mdeg (summer, daytime/ nighttime difference)	Measured
Thermal deformations of other (e.g., main reflector)	1.5 mdeg	Modeled, measured
Quasi-constant wind	2 mdeg	Modeled
Wind gusts (mechanic and servo)	3 mdeg	Modeled

Ka-band. It integrates with existing cryogenic low-noise amplifiers (LNAs) and downconverters used by ESA, without degrading downlink RF performance. It can also be used for system noise temperature measurements. In conjunction with the automated pointing error measurements, the long-term operational health of the downlink RF systems is routinely monitored with the PCS. This chapter describes the main elements of the PCS, including the radiometer developed by SED. Measured pointing data for Deep Space Antenna 2 (DSA2) are presented. Sources of pointing measurement error include atmospheric refraction especially during periods of turbulence, random noise, and gain drift in the RF systems, as well as any systematic error in the calculation of radio star calibrator position. Random error is evident in the results, and the measurement variations are calculated by PCS automatically. Systematic errors are ensured to be small by use of commercial positional astronomy software [3] and accurate system timing. Absolute pointing of the antenna is also independently verified by the excellent signal levels received when tracking spacecraft using position data calculated by external systems.

II. Systematic Pointing Error Model

Systematic pointing errors arise, for the most part, because of imperfections in the geometry of the antenna. Table 2 lists and describes the coefficients used in the systematic pointing error model (SPEM) (information taken from [1]). An example of the PCS determined value of each coefficient is also provided.

III. System Design

A block diagram of the PCS for the X/Ka-band DSA2 antenna is shown in Fig. 2. All of this equipment is designed for unattended operation and contains extensive monitor and control capability.

Table 2 SPEM coefficients

SPEM coefficient	Description	Az correction formula	Elevation correction formula	Determined value Ka RHCP, example, mdeg
IA	Azimuth (Az) encoder offset	IA		–31.8
IE	Elevation (El) encoder offset		IE	81.7
DTF	Flexure due to gravity		$DTF \cdot \cos(El)$	–80.0
AN	Tilt of azimuth axis in north direction	$AN \cdot \tan(El) \cdot \sin(Az)$	$AN \cdot \cos(Az)$	5.7
AW	Tilt of azimuth axis in west direction	$AW \cdot \tan(El) \cdot \cos(Az)$	$-AW \cdot \sin(Az)$	6.5
CA	Collimation of RF axis	$\frac{CA}{\cos(El)}$		5.9
NRX	This is a horizontal shift between the elevation axis and the azimuth axis.	NRX	$-NRX \cdot \cos(El)$	–0.04
NRY	This is a vertical shift between the elevation axis and the azimuth axis.	$-NRY \cdot \tan(El)$	$-NRY \cdot \cos(El)$	10.3
CRX1	Polarization-dependent beam squint due to the dichroic plate and elliptical mirrors.	$\frac{CRX1 \cdot \sin(Az - El)}{\cos(El)}$	$\frac{-CRX1 \cdot \cos(Az - El)}{\cos(El)}$	–0.5
CRY1	Polarization-dependent beam squint due to the dichroic plate and elliptical mirrors.	$\frac{-CRY1 \cdot \cos(Az - El)}{\cos(El)}$	$\frac{-CRY1 \cdot \sin(Az - El)}{\cos(El)}$	–0.9

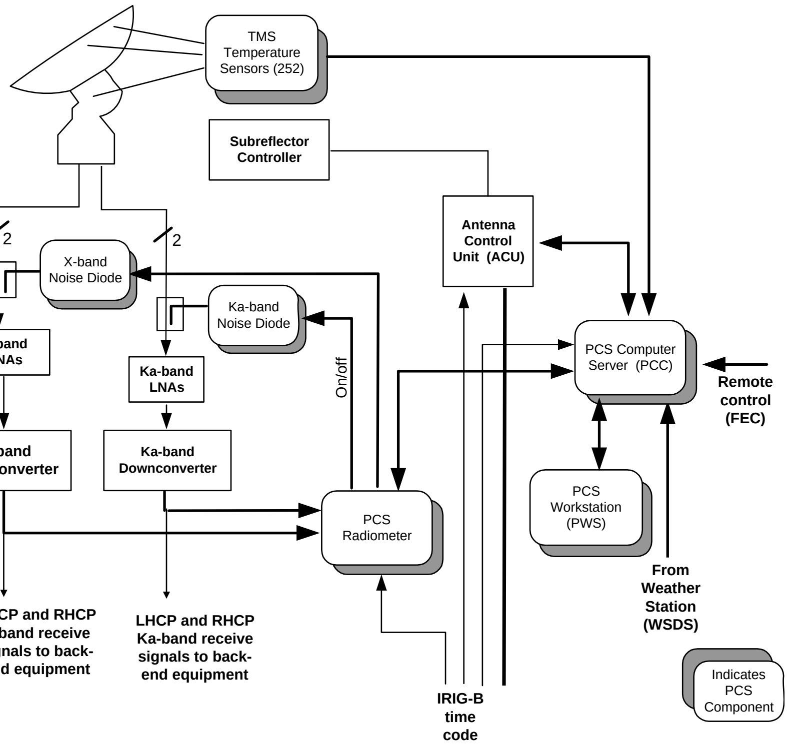


Fig. 2 PCS block diagram.

The main elements of the PCS are the following:

1) Pointing calibration computer (PCC) and its associated software. This computer runs the PCS application software that controls the pointing calibration process. It is connected via an Ethernet local area network (LAN) to ESA's mission center, for remote control, and to the site's weather station. It is connected via a separate LAN to the antenna control unit (ACU), the radiometer, and antenna physical temperature measurement system (TMS).

2) Pointing calibration workstation (PWS) and its associated software. This computer provides the local user interface used to control and monitor the operation of the PCS. A remote access capability is also provided to allow the same user interface from a remote workstation over LAN or wide area network (WAN).

3) Radiometer and its associated RF noise diodes. This equipment is used to measure downlink system noise temperature.

4) Antenna physical temperature measurement system (TMS). This consists of an array of 252 temperature sensors located on the main reflector back-structure and the subreflector quadrapod struts. Temperature data collected by this subsystem are used by the PCS to calculate the pointing error due to thermal distortion of the mechanical structure.

The PCC, ACU, and radiometer are synchronized to a common Inter-Range Instrumentation Group-B (IRIG-B) time source to ensure their actions are coordinated to a required minimum accuracy of 1 ms.

IV. Radiometer

The radiometer is a specialized noise temperature measurement system designed and manufactured at SED to provide accurate noise temperature readings during PE measurements, and pass these measurements to the PCC. The radiometer is operated in a noise adding radiometer (NAR) mode, in which a noise diode is turned On and Off in a repetitive sequence to inject a known level into the downlink during the measurement integration period. The *Y*-factor derived from the On and Off states is used to calculate *Tsys* using Eq. (1):

$$Tsys = \frac{TND}{\left(\frac{VON}{VOFF} - 1\right)} \tag{1}$$

VOFF and *VON* are the root mean square (RMS) voltages after integration by the radiometer for the noise diode On and noise diode Off portions of the measurement cycle, and *TND* is the effective noise temperature of the noise diode referenced to the input of the LNA. *TND* depends on the excess noise ratio (ENR) of the noise diode, and the insertion and coupling loss between the diode and LNA RF path. Refer to Fig. 3 for a block diagram of the radiometer including the noise diodes.

The radiometer operates over the entire downlink passband. Its integration time can be adjusted over a wide range to permit control of its noise temperature resolution. During PE measurements, the integration time is typically 1 s.

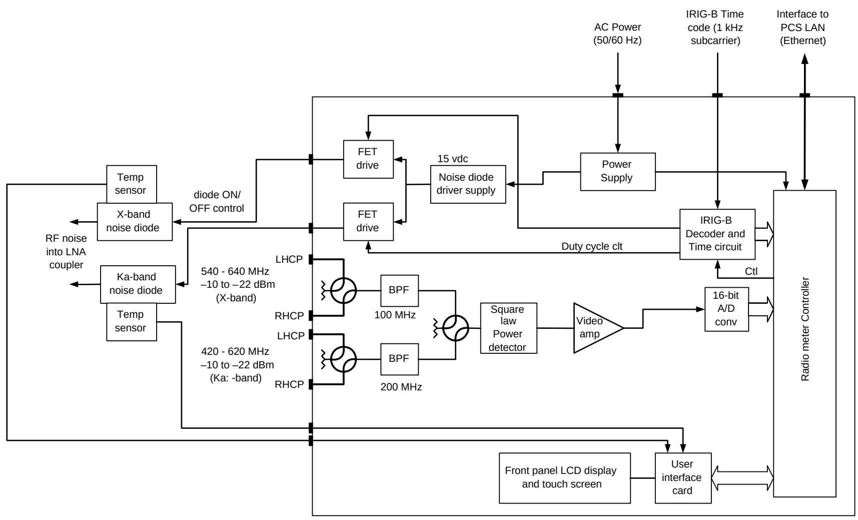


Fig. 3 Radiometer block diagram.

Table 3 Radiometer performance parameters

Description	Comment
IF input frequency band	X-band downlink: 540–640 MHz Ka-band downlink: 420–620 MHz
Number of IF inputs	Four consisting of IF inputs for: XRx Pol 1 XRx Pol 2 KaRx Pol 1 KaRx Pol 2
Integration time	Selectable in 10 ms increments up to approx 120 s. Typically 1-s integration times are used for PE measurements.
Measurement linear range	20–300 K
Absolute accuracy of *Tsys*	≤ ±5%
X-band *Tsys* short-term repeatability while scanning a radio star[a]	0.08
Ka-band *Tsys* short-term repeatability while scanning a radio star[a]	0.15 K

[a]1-s integration time, 3-sigma variation, NAR mode, calculated from PE measurement curve fit residual error, fair winter weather, all available data.

The radiometer is synchronized to the station's IRIG-B time and can be commanded to schedule multiple *Tsys* measurements at specific times. This allows the PCC to command the set of *Tsys* measurements required for a PE measurement, and ensure they are synchronized to millisecond accuracy with the current position of the antenna. Key performance parameters of the radiometer system are given in Table 3.

V. Antenna Physical Temperature Measurement System

The TMS consists of a set of 252 temperature sensors distributed on the antenna main reflector back-structure and subreflector quadrapod struts, and a data logger system to collect the temperature measurements and pass them to the PCC.

The PCC contains an algorithm that uses the temperature measurements to calculate the pointing error due to thermal distortion of the mechanical structure. The algorithm is based on finite element analysis (FEA) modeling of the antenna structure as different thermal loads are applied to different locations on the structure [2]. These are sent to the ACU as pointing corrections every 30 s. Figure 4 is a screen capture of the PWS TMS display of the antenna back-structure at 1:30 p.m. local time during the winter with clear sky conditions.

VI. Calibration System Operation

The PCS is designed with the following operating modes: 1) calibration mode and scheduler, 2) PE measurement (within a calibration), 3) SPEM calculation/transfer to ACU, 4) compensation, and 5) direct mode PE or noise temperature measurement (refer to Sec. VII). These modes are described in the following sections.

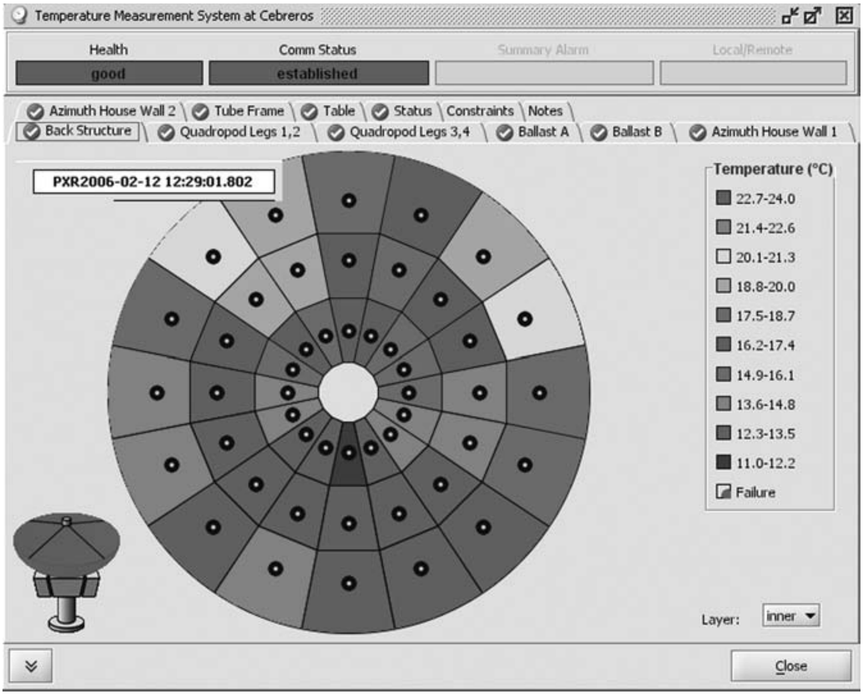


Fig. 4 Screen capture of PCS TMS display window. (See also the color figure section starting on p. 645.)

A. Calibration Mode

A typical calibration for one frequency band, one polarization takes about 8 h. In this time, the PCS can perform approximately 225 PE measurements over a wide elevation and azimuth range. It can then determine the coefficients of the SPEM model, which minimizes the error between the PE measurements and the SPEM model using the new data and optionally historical data as well. The coefficients are then passed to the ACU for use in ongoing pointing compensation. The key elements of the entire calibration process are the scheduler, the PE measurement, the SPEM calculation, and transfer to the ACU.

B. Calibration Scheduler

This is an automated planning tool used to provide an optimum measurement schedule for a calibration session. The operator first specifies a start time and duration for the session. The scheduler then selects from a library of over 70 calibration sources. These are typically quasars, whose positions are accurately known and can be calculated [3]. The sources have been selected from the very large array (VLA) database of deep space radio sources used by the radio astronomy community to calibrate the pointing of long baseline radio telescopes. The selected sources

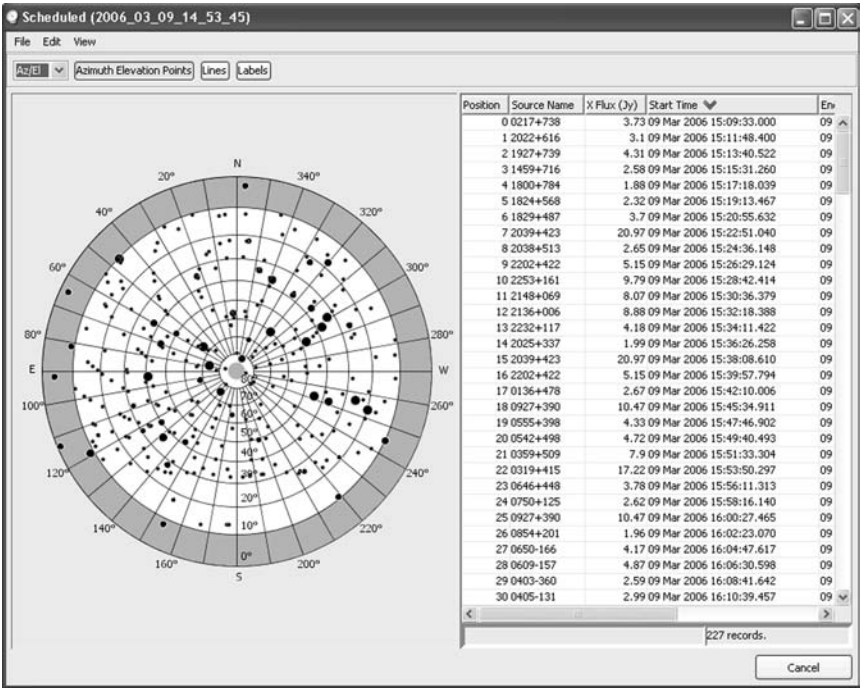


Fig. 5 Typical scheduler output (227 measurements in 8 h)—PCS screen capture.

have a very small angular extent (≤1 mdeg) relative to the beamwidth of the antenna, are separated by at least two beamwidths from nearby sources, and have flux densities ranging from 1.5 Janskies (Jy) to 20 Jy in both X- and Ka-bands.

The scheduler automatically builds a measurement schedule. Figure 5 shows a typical schedule, showing the position in az/el of measurements during the calibration session. The small dots are measurement locations while the larger dots are current star positions.

C. PE Measurement

PE measurements are made in calibration mode according to the schedule or direct mode by measuring system noise temperature with the radiometer, while scanning a radio star. The measurement algorithm uses a grid of azimuth and elevation offsets relative to the nominal position of the star. The grid is ± one 3-dB beamwidth around the nominal position of the star. For the 35-m antenna, the beamwidth is 64 mdeg at X-band, and 17 mdeg at Ka-band. Figure 6 shows a 5×5 grid. A 7×7 grid is typically used, although the PCS can accommodate grids up to 11×11.

While the PCS commands the ACU antenna to follow a trajectory through the grid points, it commands the radiometer to measure the system noise temperature

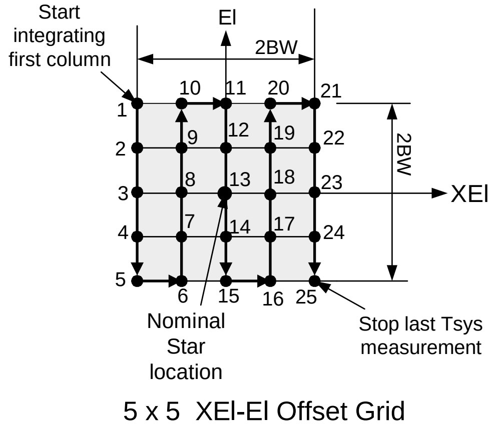


Fig. 6 PE scan grid pattern.

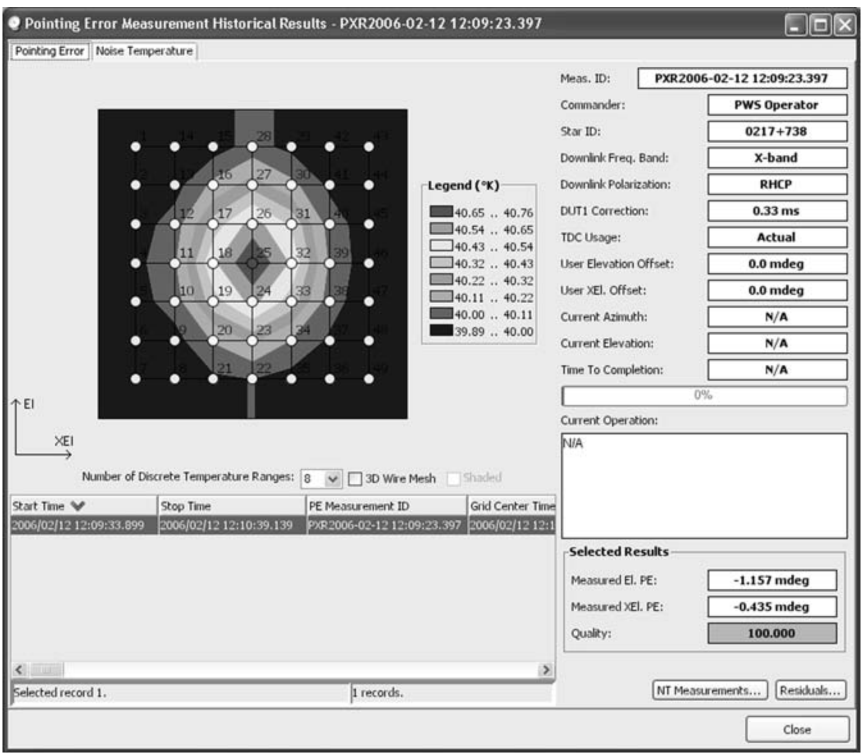


Fig. 7 PE measurement result 7 × 7 grid—PCS screen capture.

at each grid point. The radiometer integration time and the scan rate are carefully synchronized for these measurements. A measured *Tsys* profile for a 7×7 grid is shown in Fig. 7. The peak at the center of the profile is the radio star. The amount the peak is offset from the center of the grid is a direct measure of PE. Typical time for one PE measurement scan is 1 min. After the grid scan is completed, a mathematical model is fit to the measured *Tsys* data on a least-squares basis to determine the location of the RF beam relative to the nominal commanded position. The *Tsys* model used in the curve fit also provides other parameters including the peak radio star noise temperature contribution and the background temperature.

D. SPEM Calculation and Transfer to ACU

The systematic pointing error model applied by the ACU in the servo system uses 10 independent coefficients to model the systematic error of the antenna. Once sufficient (about 100) pointing error measurements are made over the whole sky, the PCS fits the multivariable SPEM model to the data to determine the coefficients. The quality of the resulting curve fit is displayed for an operator as shown in Fig. 8. This display shows the residual error between the measured data and the "best fit" model. Any remaining systematic error is immediately visible as patterns

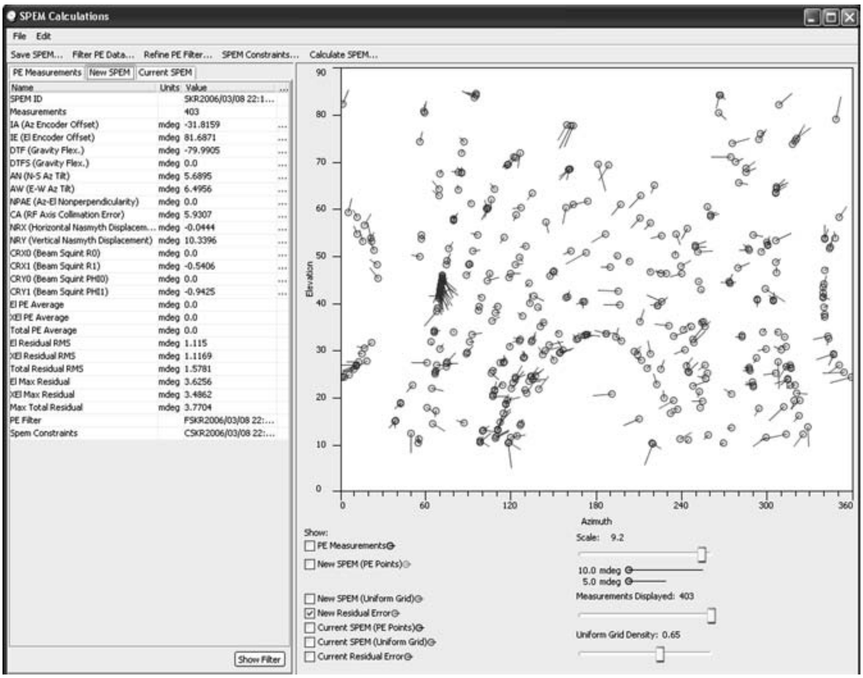


Fig. 8 Residual pointing error after SPEM coefficient calculation—PCS screen capture.

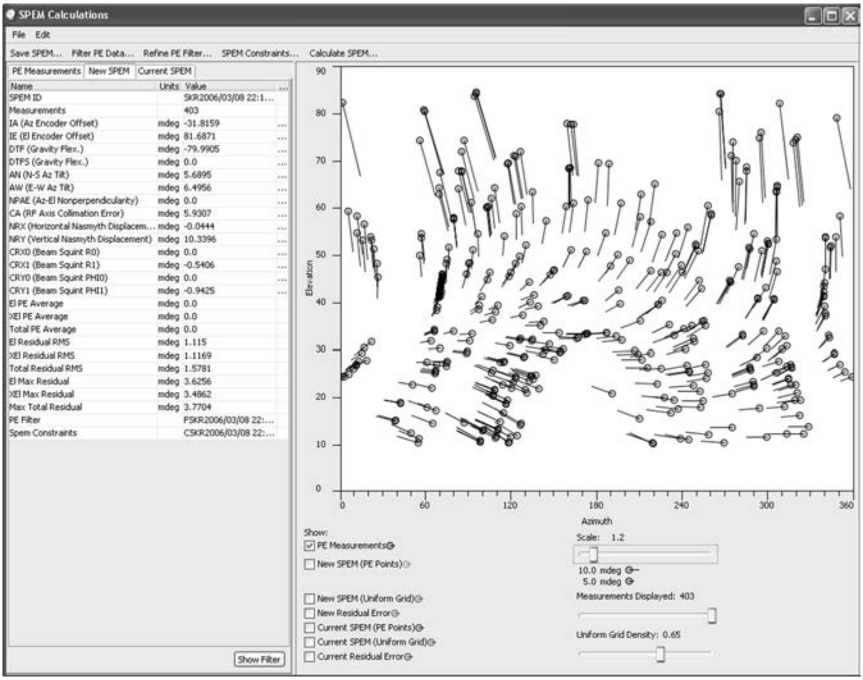


Fig. 9 Measured (raw) systematic pointing error—PCS screen capture.

in the magnitude and direction of vector residuals. The coefficients of this model are used by the servo system, after transfer from the PCS, to compensate for the systematic PE. Figure 9 is a screen capture of the PCS SPEM calculation tool that shows the raw measured pointing error as a vector field. The large elevation pointing error at high elevations is mostly due to gravity deformation. Note that the scale of the vectors in Fig. 9 is much smaller than in Fig. 8. The magnitude of the largest residual pointing error is 3.7 mdeg, while the largest measured systematic error is about 80 mdeg. This does not including any systematic error in the calculation of radio star position, although this factor is negligible due to use of proven positional astronomy software (<0.1 mdeg error) and accurate system timing.

The PCS SPEM calculation tool also gives statistics for the newly calculated SPEM in the left-hand panel of the window (refer to Figs. 8 and 9). For this particular calculation, the worst-case (3-sigma) residual pointing error is 3.77 mdeg. The RMS and average pointing errors are also given. The tool also provides a way of fixing or limiting coefficients values to a particular search range. In this example, the unused coefficients DTFS, NPAE, CRX0, and CRY0, because of redundancy with the primary set, have been fixed to zero. When the operator has completed the SPEM calculation session, the SPEM coefficients can be saved to the database. They can then be transferred directly from the PCS to the ACU across the LAN interface with the ACU in maintenance mode.

E. Compensation Mode

This is the normal operating mode of the PCS. In this case the ACU applies the SPEM, thermal distortion, and refraction corrections to commanded positions. Separate sets of SPEM coefficients are used for each frequency band and polarization according to the current polarization setting of the ACU. The system is designed such that if the PCS computer goes down, the ACU will continue to apply all corrections, except for the small thermal distortion correction. To determine refraction corrections, the ACU reads real-time atmospheric temperature, barometric pressure, and relative humidity from the site's weather station and uses this data to calculate the current refraction correction. Figure 10 shows the interaction of the PCS and external systems during compensation.

VII. Other PCS Capabilities

The PCS also includes measurement modes and offline tools used by engineering staff to quickly check antenna performance in various ways. These are described in the following sections.

A. Direct Mode PE Measurements

This allows the operator to select a single radio star for measurement with the option of continuous repeat. This has been used for investigating short-term measurement repeatability and also slower changes (e.g., due to thermal change). An example of the results for this type of test is given in Fig. 11. The smooth solid line is a trend line through the data points. The small residual pointing error, other than short-term measurement variation, could be due to residual thermal effects or other sources of error that are not fully compensated.

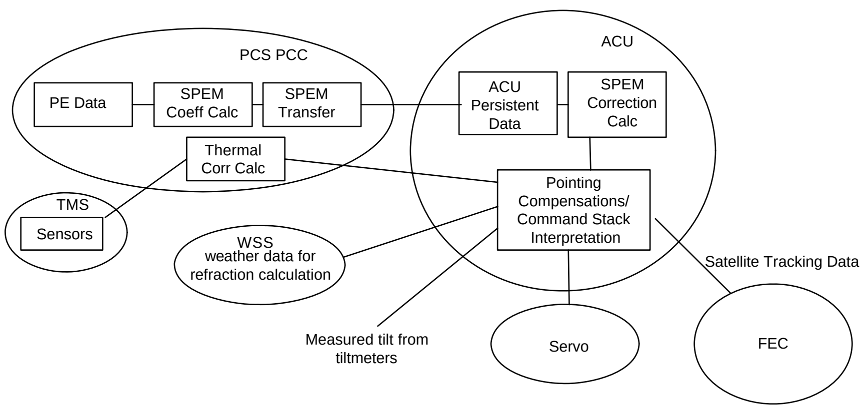


Fig. 10 System compensation mode.

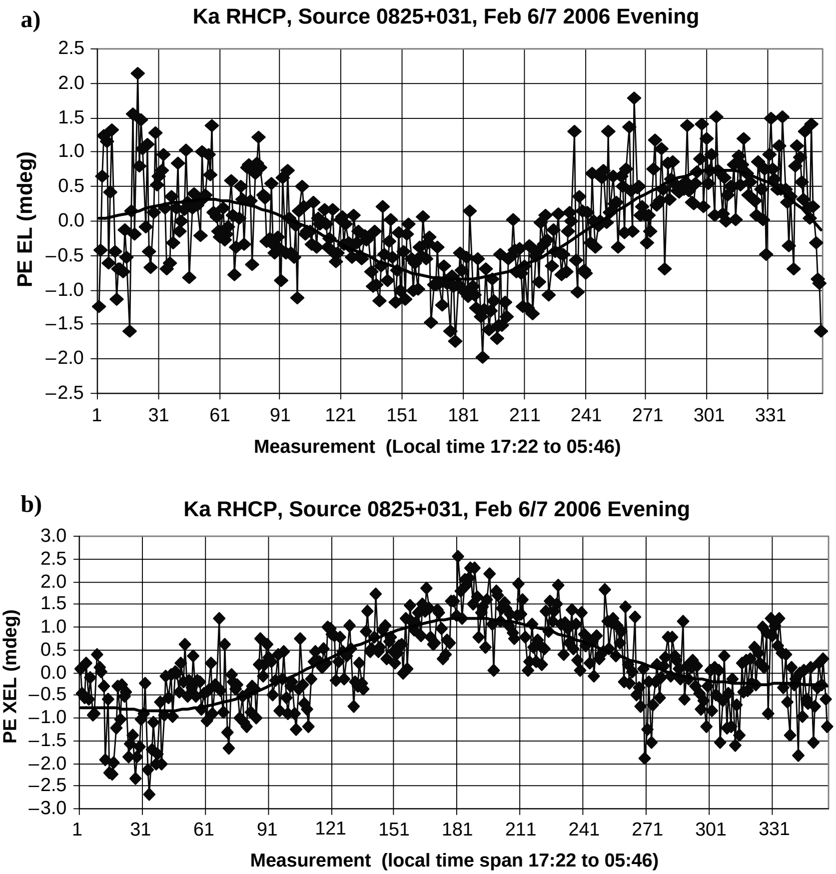


Fig. 11 PE measurements: a) elevation PE overnight and b) PE magnitude overnight XEl.

B. Direct Mode Noise Temperature Measurements

This allows engineering staff to use the radiometer via the PCS workstation to perform noise temperature measurements at specific elevation and azimuth angles. Figure 12 shows the measured system noise temperature with the antenna pointed at empty sky over a range of elevations. The noise temperature measured by a noise figure analyzer (NFA) is given for comparison.

C. Measurement Beam Squint

Beam squint causes a difference in the beam positions between polarizations due to the elliptically shaped mirrors and the phase variations of the dichroic plate in the RF path. The best-fit SPEM models from the PCS were used to compare the beam positions. For example, Figure 13 shows the contours of the measured differences in the beam position for Ka-band LHCP and RHCP in mdeg compared with

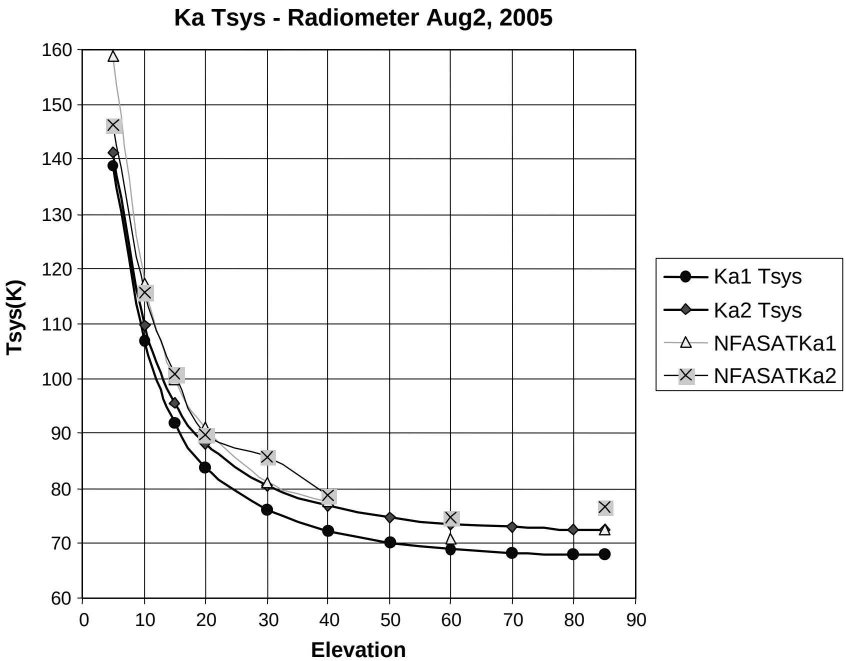


Fig. 12 Background *Tsys* vs elevation using radiometer with NFA for comparison.

the values predicted from analysis. Measured Ka-band results show a polarization-dependent pointing difference of between 1 and 2 mdeg over most of the hemisphere in general agreement with the prediction of between 1.8 and 1.9 mdeg.

VIII. Conclusion

The initial calibration and regular verification of an antenna with narrow beamwidth are critical for the operation of the ground station. SED Systems has

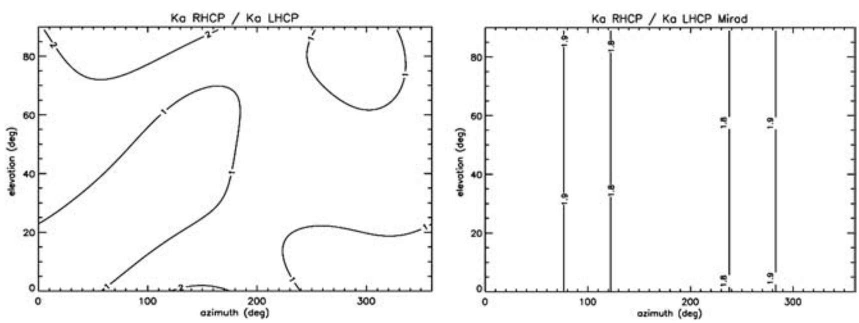


Fig. 13 Ka polarization-dependent beam pointing difference contours [measured (left) and predicted (right)].

developed, installed, and tested the PCS for the DSA2. The systematic pointing errors are determined rapidly by this system with full automation of the process. The system is ideal for use in such remotely operated, high-usage TTC antenna systems. It provides for automated planning and conduct of pointing calibration sessions, in addition to tools used to routinely check pointing accuracy and system noise temperature between calibrations. Pointing error measurement results indicate that the antenna is calibrated to point accurately with an error less than 3.7 mdeg (3-sigma variation). This includes the error of a single pointing measurement.

References

[1] Willmeroth, K., and Pausch, K., "Technical Note Systematic Pointing Error Model," Vertex Antennentechnik, GmbH, TN1003057-31440, Duisburg, Germany, Jan. 2006.

[2] Droll, P., Cozzani, A., Sieber, R., and Schäfer, M., "A Thermal Distortion Model for the ESA DSA in Perth for Compensating Thermally Induced Pointing Errors," Second ESA Workshop on Tracking, Telemetry and Command Systems for Space Applications, unpublished proceedings are compiled by ESA in cooperation with NASA and CCSDS, pp. 3.4ff, ESA/ESTEC Conference Bureau, 2200 AG Nordwijk, Netherlands, 2001.

[3] SLALIB, Positional Astronomy, Software Package, Version 134413, Tpoint Software, Abingdon, England, UK, 2003.

Chapter 13

Using Globally Connected Research and Education Networks for Space-Based Science Operations

Robert N. Bradford*

NASA Marshall Space Flight Center, Huntsville, Alabama

I. Introduction

WHAT is not common knowledge in the space-based science community is the existence of the worldwide Research and Education Network (REN) and what benefits RENs can bring to this community. In this chapter we will briefly describe what RENs are, their connectivity, underlying architecture, the services they provide, and how they can benefit space-based science. For anyone who wants to investigate RENs further to and from specific end points, go to http://www.internet2.org. From this link, a vast amount of information is available. To compare the level of services provided by RENs (in this case the Abilene Network), we will compare the network performance specifications of the NASA Integrated Services Network (NISN) and the Abilene Network. The NISN is NASA's network that provides all mission network support ranging from launches to scientific operations. Abilene is the United States' REN. It must be emphasized that the use of RENs in manned flight is not recommended at this point except in manned flight science operations as it is today. This is due to political not technical considerations. To demonstrate these benefits, we will briefly describe how the International Space Station (ISS) uses the Abilene network in its science operations and how the Japanese Solar B satellite project is planning on using RENs quite extensively in its science operations and the benefits being derived. An objective throughout this chapter is to provide adequate information for other projects to at least investigate using RENs in their science operations.

For adequate science to take place, scientists must be able to access the data from their home base at a university or scientific institute, etc. Connectivity and

*Manager, Mission Operations Laboratory.

network performance are critical at a cost that does not take away from the science being supported. In other words, if connectivity costs so much that scientific collaboration either does not happen or has a high associated cost, then clearly science looses. If data access is provided by other out-dated means, e.g., sending tapes or CDs, which has its own unique costs, not to mention their obvious inefficiencies, science again looses. A major objective of this chapter is to inform the space-based scientific community of the existence of RENs and their potential benefits.

II. REN Background

Research and education networks were conceived prior to the Internet as a Defense Advanced Research Projects Agency project. Actually the Internet as we know it today was originally formed to support the scientific community. The networking infrastructure was "taken over" as the benefits of what we now know as the Internet became clear. The first U.S. REN was the very high Broadband Network Service (vBNS) established in the early 1990s by the National Science Foundation. It was eventually replaced by the Abilene network that is the current U.S. REN. The evolution of network technology has caused a growth explosion in RENs worldwide. Now the world is connected via RENs. The only continent not readily connected is Africa, and even this is changing. Within most countries there exists a national REN. These national RENs provide connectivity to colleges, universities, science centers, institutes, and governmental agencies within a country. The GÉANT (the European REN), Asia Pacific Advanced Network (APAN), and America's Pathway (AMPATH) organizations provide intercontinental connectivity between the United States and Europe, Asia and Oceania and the Caribbean, South/Latin America, respectively. These are the connector links traversing continents and oceans.

One of the problems in space-based science is getting the science (data) from the source, a satellite, to a satellite receiving station, processed at a science center and out to the science community at a university or institute and on to other locations for possible scientific collaboration. A break in this line, generally due to cost, negates effective collaboration, especially internationally. If a scientist must wait long periods of time to get access (via tapes or CDs), or the data are corrupted and must be resent again, then scientific collaboration starts to break down. Today very large data sets are being created that require high bandwidth networks.

To illustrate how a space-based scientific endeavor can benefit from RENs, we will describe how the International Space Station uses Abilene for onboard scientific operations and how the Solar B Satellite Project will use RENs when it is launched.

III. Discussion of Research and Education Networks

In this section we will attempt to educate the reader on the very basics of RENs. This section is not all inclusive, and we encourage the reader to visit the individual web sites associated with a specific country's network connectivity. Of significant importance is the lack of nationalism associated with RENs. There is a sense of

duty that seems to preclude nationalistic tendencies where the networks come first. This is an empirical observance by the author not founded by any research.

A. Purpose

The purpose of RENs is straightforward. They exist to provide network connectivity for research and educational purposes. They do not support the commodity Internet. Research covers a range from network research to discipline specific scientific research. Education covers all aspects of education, including access to information, streaming video, collaborations, and online teaching. Ancillary support is generally allowed that is related to commodity Internet traffic, e.g., voice over Internet protocol (VoIP), access to the Internet for research purposes, and e-mail.

B. REN Organization

Each international region and individual nations have their own network organization infrastructure. The Asian Pacific Advanced Network and GEANT are examples. For network-specific discussions in this chapter because RENs are so prevalent, we will discuss in more detail the United States' Abilene REN. The Abilene network is supported by the University Corporation for Advanced Internet Development (UCAID) under Internet2, a consortium of over 205 nationally recognized colleges and universities. UCAID is a private nonprofit corporation that provides the oversight, funding, and management for the Abilene network. Funding is provided from many sources. Schools and other organizations must join and pay dues to connect to Abilene. Grants and research funding comes from the National Science Foundation, which is a funding source for Abilene. The last major funding sources are the sponsoring commercial networking and governmental science organizations.

There is no worldwide recognized REN control authority. The Internet2 does provide a global centralized network operations center for international networks and connectors. The Global Network Operations Center (NOC) is located at the Indiana University/Purdue University campus at Indianapolis (IU/PUI). This Global NOC, however, does not provide service within a national network and does not have insight into internal operations.

C. Condition of Use Policies

Most, if not all, RENs have a condition of use policy. The overall policies concerning use of RENs are different depending on region and national locations, but generally the use must be related to scientific and network research and education. Membership restrictions vary between national and international entities. However, when traversing other networks to get to a faraway end point, whether traversing it or delivering to it, it is not required that memberships to all networks, in between end points, is required. Once a membership is accepted by one entity, it is generally recognized by all other RENs including connectors.

The Abilene Network states: "As a project of Internet2, the Abilene network has established Conditions of Use (CoU) that seek to advance the Internet2 project's goal of encouraging and enabling the development of advanced network

applications. Abilene provides high-performance networking for data traffic among participating gigaPoPs and Regular Members, as well as other organizations whose connectivity benefits higher education in the United States. Abilene traffic primarily and clearly serves the teaching, learning, research, and clinical missions of US higher education, plus related support infrastructure, services, and content. Abilene does not seek to compete with the commodity Internet or other telecommunications services, and is not intended to carry any commercial traffic unrelated to Internet2 goals, or any traffic with proprietary, classified, or illegal purposes. All Abilene participants agree to comply with conditions and charges set by Internet2 for using the Abilene network" ([1] abilene.internet2.edu/policies/cou.html).

As stated in the Abilene's CoU, scientific data including scientific space-based operations is included. The International Space Station remote scientist uses Abilene to conduct all ISS science, including commanding of experiments, receipt of telemetry, voice, and streaming video.

D. Current Services Provided by RENs

The span of network technology advancement within RENs is very location dependent. Network infrastructure varies greatly. The installed infrastructure of the Abilene network and its connection points are all based on advanced fiber technologies. It is not uncommon to see T1s and T3s in some of the more remote locations of the world. The typical services provided are network operations center support for operations, Internet Protocol version 6 (IPv6), multicast and quality of service (QOS). Abilene offers a redundancy service at their connection points, which eliminates single points of failure and increases reliability. These technologies are implemented at varying degrees worldwide.

E. Brief Connectivity Overview

The Abilene REN architecture consists of high-speed fiber optics, routers with a significant ability to reroute when failures do occur, peering and gigaPoPs (high speed points of presence) between abilene and regional networks. Figure. 1 depicts the topology of the Abilene network. The network is comprised of rings that provide significant redundancy and high reliability. During Hurricane Katrina the link between Atlanta and Houston, which runs through New Orleans, was disabled. Even though the failure was catastrophic, no reduction in service was encountered by users. The service was restored 9.5 days after the storm. There is no location on the network that is not serviced by at least two different links. In the event of one link going down, the traffic is immediately rerouted on the other link to maintain service. As Table 1 indicates, sufficient bandwidth exists on all links to handle this type of rerouting.

Because of the vast interconnectivity between the international RENs, they act as one very large ring-based network worldwide. The international REN literally traverses the world, and goes as follows: starting at the StarLight connector in Chicago, to the Asia Pacific Advanced Network to Japan, to Gloriad across Russia, to Europe, where it connects with GEANT across the Atlantic back to StarLight. Although this description sounds like a single circuit, this connectivity

Fig. 1 The Abilene network backbone ([1] abilene.internet2.edu/maps-lists).

is made up of multiple 10 Gbps fiber optic circuits, connector points, and national RENs. A break in any one of these circuits does not materially affect overall service.

Table 1 Abilene network typical utilization for the week of 3–9 April 2006[a]

Date	Direction	Average Mbps	% Utilization	Total Xfers in Gbytes
9 April	In	22,910.00	2.727	241,629
	Out	23,108.40	2.751	243,722
8 April	In	24,191.40	2.88	255,143
	Out	24,386.50	2.903	257,201
7 April	In	27,577.60	3.283	290,858
	Out	27,729.00	3.301	292,454
6 April	In	27,397.70	3.261	288,960
	Out	27,583.40	3.284	290,918
5 April	In	27,955.50	3.328	294,843
	Out	28,139.90	3.35	296,788
4 April	In	29,024.70	3.455	306,120
	Out	29,125.70	3.467	307,185
3 April	In	27,355.10	3.256	288,511
	Out	27,538.10	3.278	290,441
Total In		26,630.30	3.17	280,866
Total Out		26,801.60	3.191	282,673

[a][5] stryper.uits.iu.edu/abilene/aggregate/html/.

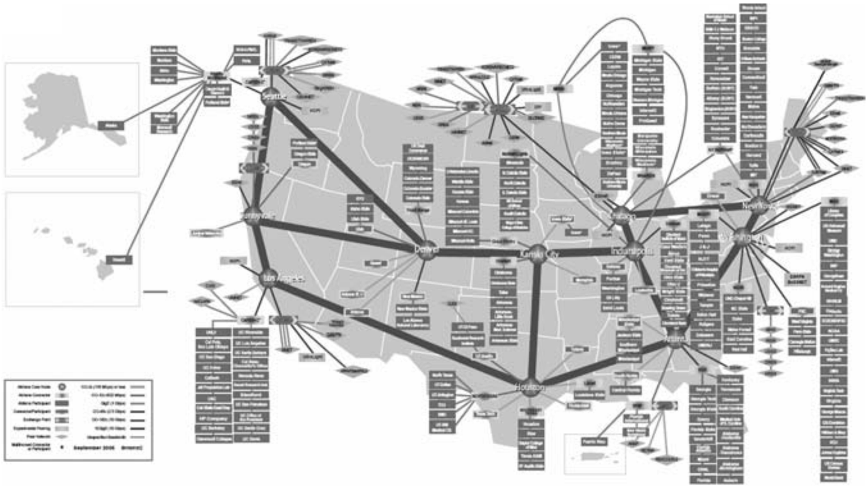


Fig. 2 The Abilene network logical map ([1] abilene.internet2.edu/files/abilene-logical-map.pdf).

Shown in Fig. 2 and 3 is the logical view of the Abilene network showing the connecting regional and state RENs and international RENs, respectively, and shows how extensive the connectivity is between end points. What is not shown is the vast connectivity at the local level within a state ending in classrooms and laboratories. There are 36 Abilene connectors located nationwide.

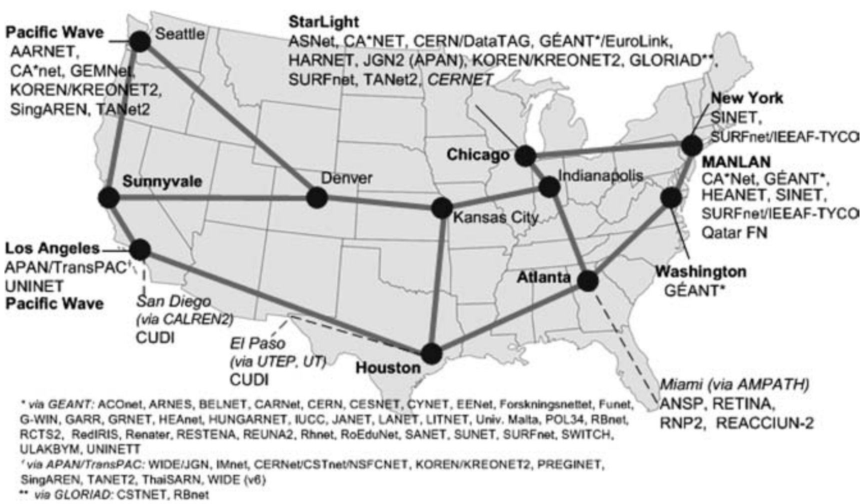


Fig. 3 The Abilene network connector networks ([1] abilene.internet2.edu/maps-lists/).

F. Peering Relationships and Connectors

What makes the REN networks so powerful is the connectors that link them together at the state, regional, national, and international levels. The connectors to Abilene in Fig. 3 show the Abilene connectors, where an OC3 is 155 Mbps, OC12 is 622 Mbps, OC48 is 2.4 Gbps, and OC192 is 10 Gbps. These types of connectors are typical in the industrialized world and many up-and-coming nations worldwide. What make the worldwide REN so robust are the peering relationships between regional and national RENs and these relationships with transoceanic networks like TransPAC for APAN.

Almost without exception networks have peering relationships with multiple networks and multiple peering locations with the same network. For example, between Abilene and the NISN there are three different peering locations, one on the west coast, one on the east coast, and one in Chicago. These multiple peering locations will avoid single points of failure because a peering relationship equates to a physical location.

A peering relationship is an agreement between two networks to transfer traffic between themselves and to provide a physical place to accomplish this transfer. These peering locations are called various things like gigaPoPS, connectors and possess specific names like Pacific Wave and StarLight. For the United States there are major peering locations in New York City, Miami, Chicago, Seattle, and Los Angeles. Their location somewhat dictates the emphasis of their connection points, e.g., east coast to Europe, west coast to Asia, and Oceania and Southern Gulf Coast to the Caribbean, Latin, and South America. Without peering relationships there would be no RENs or for that matter no Internet.

G. Specific Network Operational Specifications for Space Operations

For space-based satellite operations, National Aeronautics and Space Administration (NASA) uses the NASA Integrated Services Network (NISN) to provide all network services. NISN has four levels of service. They are standard IP

Table 2 NASA Integrated Services Network (NISN) Internet protocol performance specifications[a]

Service category	Availability [3], %	Restoral time [3]	Coverage period	Acceptable packet loss, %	Round-trip time [6]
Real-time critical	99.98	<1 min [5]	24 × 7	0.001	<120 ms
Mission critical	99.95	2 h [4]	24 × 7	0.001	<120 ms
Premium	99.50	4 h [4]	24 × 7	<1.0	<100 ms
Standard	99.50	<24 h [2,4]	6 a.m. Eastern Monday to 6 p.m. Pacific, Friday	1.0	<250 ms

[a][4] nisn.msfc.nasa.gov/DocumentPages/Services.html.

(SIP), premium IP (PIP), mission critical IP (MCIP), and real-time (RT) mission critical. Table 2 provides the performance specifications for each service taken from the "NISN Services Document" [4]. The following is a brief description of each service.

1) SIP: This service provides for basic data networking connectivity using the IP suite. SIP service is the commodity Internet service that provides the Agency's link to the Internet in general. It provides basic universal Internet connectivity with minimal performance guarantees or restrictions on acceptable use. SIP service is open to the public to enable access to publicly available NASA information sources such as World Wide Web services.

2) PIP: This service provides a premium level of data networking connectivity using the IP suite. PIP service is differentiated from SIP service in that it provides a higher performance level, higher priority for problem resolution, and is not directly connected to the general Internet. PIP connectivity to the general Internet is through a controlled gateway and is implemented on an exception basis only. PIP service is most appropriate for internal Agency networking requirements where the Agency's operations should be isolated from the general Internet. PIP service is not used in space flight operations. It is used in space flight science operations for ISS.

3) Mission Critical IP: This service provides a mission critical level of data networking connectivity using the IP suite with controlled access and security measures. MCIP service is differentiated from SIP service in that it is engineered as a closed system to support space flight mission critical telemetry and data flows. All systems and facilities connected to the MCIP service shall meet the specified IT security level. Access to and from the general Internet and other NASA IP services is extremely limited and provided on a strictly managed "by exception" basis. MCIP service is most appropriate for critical space flight mission support data and telemetry flows that require 1) an extremely high level of availability for mission success and 2) no general Internet access.

4) RT Mission Critical: This service provides a real-time critical level of data networking connectivity with emphasis on meeting real-time telemetry transport using the Internet protocol suite. Real-time critical IP (RCIP) service is primarily differentiated from mission critical IP (MCIP) service in that it is engineered with a higher level of redundancy to achieve the added level of availability. This service employs the same security and connectivity features and limitations as the mission critical service. It is used to support life and vehicle threatening activities where no disruption of network service can be tolerated and delivery is in real-time less the laws of physics ([4] http://nisn.msfc.nasa.gov/DocumentPages/Services.html, April 2005).

Generally speaking, security is embedded in the various services, and bandwidth sharing is not provided.

1. *Mission Critical Networks (NISN Provided)*

Table 2 provides the specification for each service level as published by NISN. The following should be noted:

1) A capability for immediately switching to an alternate data path shall exist.

2) These restoral times represent the time to restore service to the user and assume immediate access to the user's facility to repair/replace equipment if necessary.

3) The 24-h restoral time results from the decreased priority given to standard service as compared to the other classes of service and from the fact that standard routed data service equipment is often a considerable distance from a NASA operating location.

4) These values apply only for those parts of the Wide Area Network (WAN) service supported by the NISN mission services backbone infrastructure. These values do not apply to tail circuits unless the circuits/services were specifically ordered and supplied with diverse routing end-to-end.

5) Round-trip time (latency) is specified for data flow between WAN nodes controlled and operated by NISN. Latency is a function of distance and carrier capabilities. User applications that are sensitive to latency shall be engineered to account for the upper limit round-trip times specified in Table 2.

2. Better Than Best Effort (REN Provided)

The statistics in Table 1 are taken from the seven-day period starting 9 March 2006, and show the total amount of data sent across the Abilene network. These data are archived at the Internet2 Abilene Network Operation Center. As shown, the aggregate traffic on Abilene is significant while the percentage of use is low. While this does not reflect what may be happening over a greater time frame, it is indicative of the bandwidth that might be available to science and education. Although this may appear as over-provisioning, it is actually reflective of the fiber technology used.

3. Availability

Operational specifications vary widely according to the network. For this chapter we will investigate the Abilene network in terms of availability, packet loss, and latency. It must be recognized that any look into performance is only a snapshot. To accomplish this, the week of 3–9 April 2006, and the preceding year has been selected because schools are in session and no holidays are near.

Availability for Abilene was 99.99791% for the year ending 9 April 2006. There has never been a failure that has resulted in a loss of availability to the Abilene user community. When failures occur, they are measured in milliseconds while traffic is rerouted usually without packet loss occurring. As mentioned, when Hurricane Katrina took out the Atlanta to Houston link, no loss of service occurred as traffic was rerouted over other links to traverse east and west. Since the southern route was broken, making the Kansas City to Denver circuit a single point of failure, during the outage, a circuit was “borrowed” from the National Lambda Rail from Seatle to Chicago, thus establishing another path until the Houston to Atlanta service was reestablished. Availability of regional networks is similar to Abilene. However, some state RENs during school hours have congestion problems that limit their performance.

4. Latency

The Abilene network latency is measured one way for all nodes to all other nodes. Since Abilene is a fiber-based network, the latency between nodes is only

limited by physics and is essentially the measured speed of light between nodes plus routing determinations along the way.

5. *Packet Loss*

The Abilene network essentially operates without any packet loss. In general, there is a relationship between packet loss and errors per second in that if there are no errors per second there follows that there is no packet loss occurring. As Table 3, shows Abilene experienced virtually no packet loss during calendar year 2005. The packet loss value in the table for all receivers was zero. The only time where loss did occur was when the circuit between Atlanta and Houston, which goes through New Orleans, was down for 9.5 days and was measured in the 10s of packets not adequately significant to measure on in Table 3.

H. NISN to Abilene Comparison

To demonstrate the adequacy of using RENs for space and science operations, a comparison of the performance specification between NISN and Abilene is required. The following are specifications for NISN and measurements for Abilene within the respective network backbones. In Table 4 is a comparison of the actual performance of the Abilene network and the published performance specifications of the NISN network. NISN actual performance statistics are not available. Because NISN measured performance statistics are not available for this chapter, it is assumed they are within the specification. As can be observed, the Abilene network availability exceeds all NISN categories including real-time mission critical. Restoral times exceeded the NISN mission critical. It should be pointed out that restoral periods do not equate to loss of service, when restoral is defined as restoration of lost service and is measured in milliseconds not minutes or hours to a user. The IU/PUI Network Operations Center operates 24 days, 7 days a week. The NISN packet loss for real-time mission critical of 0.001% is exceeded by Abilene's 0% packet loss performance.

In an attempt to quantify the NISN specification compared to the Abilene performance statistics for the week of 3–9 April 2006, a comparison is presented in Table 4 based on the traffic presented in Table 1 for the week of 3–9 April 2006. What is presented is what would be allowed under the NISN real-time mission critical service in effect as of May 2006. Packet loss is based on 563,539 Gbytes of traffic in and out of Abilene. The availability is based on a one-week timeframe. The Abilene latency is based on the worst case for calendar year 2005 for traffic between New York City and Los Angeles.

I. REN Security

Security of the network itself is the responsibility of the individual RENs. Denial-of-service attacks do occur at the end networks, and hackers do at times use the Abilene network to conduct their attacks. That said, the NOC and Global

Table 3 Abilene latency (in ms)/packet loss statistics (in %) for IPv6 traffic in calendar year 2005[a]

Receivers	Atlanta	Chicago	Denver	Houston	Indianapolis	Kansas City	Los Angeles	New York City	Sunvale	Seattle	Washington, D.C.
Atlanta	0.023/ 0.00	7.645/ 0.00	15.680/ 0.00	9.995/ 0.00	5.750/0.00	10.356/ 0.00	27.090/ 0.00	10.199/ 0.00	28.035/ 0.00	28.376/ 0.00	8.065/0.00
Chicago	7.896/ 0.00	0.023/ 0.00	12.428/ 0.00	14.299/ 0.00	2.011/0.00	6.624/ 0.00	28.785/ 0.00	10.109/ 0.00	24.300/ 0.00	24.641/ 0.00	12.238/0.00
Denver	15.589/ 0.00	13.052/ 0.00	0.023/ 0.00	13.542/ 0.00	10.311/0.00	5.661/ 0.00	17.241/ 0.00	23.073/ 0.00	13.615/ 0.00	13.954/ 0.00	25.201/0.00
Houston	17.950/ 0.00	14.282/ 0.00	13.308/ 0.00	0.023/ 0.00	12.392/0.00	7.773/ 0.00	16.293/ 0.00	24.298/ 0.00	19.665/ 0.00	27.957/ 0.00	26.427/0.00
Indianapolis	5.951/ 0.00	329.655/ 0.00	12.877/ 0.00	12.374/ 0.00	0.023/0.00	4.698/ 0.00	26.524/ 0.00	11.998/ 0.00	22.374/ 0.00	22.714/ 0.00	14.127/0.00
Kansas City	11.092/ 0.00	6.602/ 0.00	5.407/ 0.00	7.765/ 0.00	4.709/0.00	0.023/ 0.00	21.826/ 0.00	16.619/ 0.00	17.769/ 0.00	18.110/ 0.00	18.744/0.00
Los Angeles	26.282/ 0.00	28.602/ 0.00	16.758/ 0.00	16.460/ 0.00	26.708/0.00	22.095/ 0.00	0.023/ 0.00	38.619/ 0.00	4.418/ 0.00	12.705/ 0.00	34.167/0.00
New York City	10.428/ 0.00	10.119/ 0.00	21.980/ 0.00	19.926/ 0.00	12.037/0.00	16.647/ 0.00	38.421/ 0.00	0.023/ 0.00	34.325/ 0.00	34.664/ 0.00	2.221/0.00
Sunvale	28.240/ 0.00	352.237/ 0.00	12.465/ 0.00	19.614/ 0.00	22.376/0.00	17.762/ 0.00	4.240/ 0.00	34.285/ 0.00	0.022/ 0.00	8.368/ 0.00	36.416/0.00
Seattle	28.461/ 0.00	24.790/ 0.00	13.753/ 0.00	27.966/ 0.00	22.893/0.00	18.278/ 0.00	12.565/ 0.00	34.802/ 0.00	8.442/ 0.00	0.024/ 0.00	36.935/0.00
Washington, D.C.	8.295/ 0.00	12.251/ 0.00	24.086/ 0.00	17.787/ 0.00	14.167/0.00	18.863/ 0.00	33.967/ 0.00	2.225/ 0.00	36.455/ 0.00	36.796/ 0.00	0.023/0.0

[a][5] ndb1-lmt.abilene.ucaid.edu/ami/owamp_status.cgi/14231676561123180544_14411656736165330944.

Table 4 Abilene network and NISN performance comparison

	NISN	Abilene[a]	NISN	Abilene[a]	NISN	Abilene[a]	NISN	Abilene[a]	NISN	Abilene[a]
Service category	Availability [3], %		Restoral time [3]	All restorals within 2 h with no loss of service to users[b]	Coverage period	IU NOC is 24×7	Acceptable packet loss, %	0 Packet Loss	Round-trip time [4]	Between nodes
Real-time critical	99.98	99.9979	<1 min [6]		24×7		0.001	0	<120 ms	Worst case for the week of 4/9/06: 38 ms one way
Mission critical	99.95		2 hr [4]		24×7		0.001		<120 ms	
Premium	99.5		4 hr [4]		24×7		<1.0		<100 ms	
Standard	99.5		<24 hr [2,4]		6 a.m. Eastern, Monday to 6 p.m. Pacific, Friday		1		<250 ms	

[a]Actual performance. [b]Except Hurricane Katrina (9.5 day restoral time). [c]Between Abilene backbone nodes. http://globalnoc.grnoc.iu.edu/weekly_reports/abilene/recent_outages/20060409.htm.

Table 5 Quantifying the NISN performance specification against the Abilene performance for the week of 3–9 April 2006

Performance category	NISN Allowable	Abilene Actual	Measurement
Packet loss	54,789	0	Packets for the week
Service interruptions	120.96	0[a]	Seconds for the week
Latency	120	76[b]	milliseconds round trip

[a]Not added are the few milliseconds associated with re-routing.
[b]Worst case between New York City and Los Angeles during CY05.

NOC are on a constant vigil and work with the responsible entities to combat these types of attacks. As with most aspects of security, the Abilene network does not divulge specific security mechanisms that are in place.

Security for user data is the responsibility of the user.

IV. Future Technologies and Bandwidth Availability

For potential science users a question should be asked: What about the future? Generally speaking, bandwidth in networking is becoming a non-issue. In the future dense wave division multiplexing (DWDM) will be serving up almost limitless bandwidth. Of course, as bandwidth becomes available, so does the applications to eat it up! Currently DWDM-based technologies are installed at the trunk level within a networks infrastructure. Light-based switching and routing will allow faster and more reliable distribution including distribution to the desktop. Fiber to the desktop will enable desktop ingest speeds unheard of today. Network technologies are evolving rapidly and should provide adequate bandwidth for years to come.

V. Brief Overview of ISS REN Use and the Solar B Satellite Planned Use

The International Space Station's Payload Operations Integration Center (POIC) located at the Marshall Space Flight Center in Huntsville, Alabama, is the focal point for all ISS-based science operations. Since November 2001, the POIC has coordinated all ISS scientific activities. This includes receipt of telemetry from ISS and distributing it to the various principle investigators throughout the United State. Also, all voice operations between the ISS crew and PIs are conducted through the POIC. Downlink video from ISS in the form of two 4 Mbps streams is received via multicast from the Johnson Space Center to Marshall Space Flight Center (MSFC) and PIs. PIs command their instruments via the POIC. Most POIC to PI operations is conducted over the Abilene network.

"Solar-B is the follow-up mission to the very successful Japan/UK/US Yohkoh mission. Using a combination of optical, Extreme Ultraviolet and X-ray

instrumentation Solar-B will study the interaction between the Sun's magnetic field and its corona to increase our understanding of the causes of solar variability. It is due for launch in September 2006. Included in Solar-B's instrumentation is a 0.5 m optical telescope, an EUV imaging spectrometer and an X-ray/EUV telescope. The instruments will work together as an observatory" ([6] www.mssl.ucl.ac.uk/ www_solar/solarB).

The Solar B network requirements are divided into two categories, immediate post-launch and then the ongoing science operations. The first category is to provide network support for satellite test and checkout just after launch. During the initial satellite checkout period, scheduled for up to 20 days after launch, data from the satellite will be received at ground receiving stations at Santiago, Chile, Wallops Island, Maryland, and several Japanese ground stations. The plan is to receive real-time flows from Santiago and Wallops through Goddard Space Flight Center that contains engineering telemetry necessary to assess spacecraft conditions and to make any necessary adjustments. These flows will be routed to MSFC, where they will be available by FTP to Japan through existing Huntsville Operation Support Center (HOSC) systems.

Once the satellite has been checked out and is operational, the x-band originated telemetry will flow from satellite receiving stations in Svalbard, Norway, and Japan to the Solar B control center in Sagamihara, Japan. This flow will be transmitted several times per day for the life of the satellite. Once received at the Solar B control center, the data will be decommutated (separated by instrument), and the instrument-specific data will be sent to three locations in the United States, one in Britain and one in Norway for archiving and access by scientists. Also, a database will be maintained in Japan. Once recorded/archived at one of these locations, this science data can be accessed by anyone authorized to receive it. It is estimated that up to 30 Gb of telemetry will be transmitted daily to Japan from the satellite and comparable flows to the archiving locations and from the archiving locations to the scientific community.

VI. Conclusion

The benefits of REN use are many. Although not addressed in this chapter, national and international RENs, especially in the industrialized world, perform equal to or exceed the space operations network specifications required to support even critical space operations. The overall cost is minimal. The connectivity is everywhere and growing. The underlying technology is cutting edge but not bleeding edge and allows more bandwidth availability than demand.

The performance of the Abilene network and generally of national and international RENs is more than adequate to support space-based science operations for science conducted on manned and unmanned vehicles and satellites. It meets or exceeds the NISN published performance specifications for real-time mission operations. The worst-case one-way latency far exceeds the NISN specification, e.g., 38 ms worst-case one-way latency between New York City and Los Angeles for the week of 3 April 2005. Abilene virtually experiences no packet loss and for one year prior to the April 3 week had a 99.997% availability with no interruptions of service. The network possesses adequate bandwidth based on utilization

statistics and its technology base. It uses advanced networking technologies while continuing to plan for the future.

It needs to be emphasized that using RENs does not replace use of NISN mission-critical services where loss of network services equates to loss of life or mission, e.g., loss of a satellite. This is as much due to politics rather than technology and loss of control whether perceived or actual. Losing some science data that are generally recoverable vs losing a life or spacecraft requires varying levels of control. However, since all network traffic is transported on the same media regardless of whether it is internet or spacecraft control data, there is room to debate the effectiveness of separate mission networks. As already mentioned, RENs do not replace commercial network services but are in place to support research and education. Quite simply, RENs can provide the research data to end users and on most RENs to government research institutes and governmental RENs. Access to data by the research community (regardless of discipline) is essential to scientific collaborations. As shown, connectivity is worldwide, with stellar performance that enables collaboration at many levels. Voice- and video-conferencing, data sharing, and access to analysis and scientific results are transmitted and received instantaneously all without the need to purchase network services or circuits, which avoids the cost and resources that would be required to procure and operate individual networks/circuits. These costs would be significant, especially if every scientific collaboration was required to implement dedicated network services. Every dollar saved is a dollar that can be applied to the scientific endeavor. As seen, the network performance of Abilene is more than adequate to support virtually any type of space-based scientific endeavor.

However, there are a few minor drawbacks against REN use. Specifically, the lack of control most government agencies crave is not present. However, there is a significant amount of cooperation between networks. No centralized end-to-end REN control authority exists, although significant cooperation exists between national, regional, and intercontinental RENs. Finally, support could dry up if funding became difficult due to international politics.

A word of caution: be careful not to spoil the REN opportunity by abusing it. Do not use it for purposes other than those that have been published in the Conditions of Use.

Acknowledgments

The author would like to acknowledge the contributions of Eric Boyd, Rich Carlson, Matt Zekauskas, and Jim Williams of Internet2 and the Indiana University/Purdue University at Indianapolis Network Operations Center for providing data of Abilene performance, pointing the author in the right direction.

References

[1] "Internet2: Building Tomorrow's Internet," Ann Arbor, MI, 2006, http://abilene.internet2.edu/policies/cou.html [retrieved 8 Feb. 2006].

[2] "GEANT Topology," Cambridge, UK, 2006, www.geant.net [retrieved 12 April 2006].

[3] "Asia Pacific Advanced Network Topology," Bangkok, Thailand, 2006, www.apan.net/documents/network-topology-8.1.ppt [retrieved 8 Feb. 2006].
[4] "NISN Services Document," NASA Marshall Space Flight Center, Huntsville, AL, 2005, http://nisn.msfc.nasa.gov [retrieved 16 May 2006].
[5] "Indiana University Abilene Network Operations Center, Network Monitoring," *Network Monitoring Databases* [online databases], http://ndb11mt.abilene.ucaid.edu/ami/owamp_status.cgi/14231676561123180544_14411656736165330944.
[6] "An ISAS/NASA/PPARC Mission to Study the Sun," Tokyo, Japan, 2006, www.mssl.ucl.ac.uk/www_solar/solarB [retrieved 9 Feb. 2006].

Chapter 14

Migration of a Mission Control System to an Upgraded Infrastructure

Daniel Ponz,* Mauro Casale,† George Kerr,‡ and Dietmar Heger§
European Space Agency, Darmstadt, Germany

I. Introduction

LAUNCHED in December 1999, the X-Ray Multi-Mirror Mission (XMM), XMM-Newton, is the second cornerstone mission of the Horizon 2000 program of ESA. In December 2005, the Science Program Committee extended operations of the mission until 31 March 2010. The decision was largely based on the outstanding science results, with over 1000 scientific papers published. The spacecraft is a high throughput X-ray telescope equipped with three X-ray imaging cameras, two X-ray spectrometers, and an optical telescope. Spacecraft control operations are performed in real time from the Mission Operations Center (MOC) at the European Space Operations Centre (ESOC) in Darmstadt, Germany, via a network of ground stations located around the Earth, that provide continuous coverage of the spacecraft orbit. Instrument operations and user support are performed from the Science Operations Centre (SOC) at the European Space Astronomy Centre (ESAC) in Villafranca del Castillo, Spain.

Because of the excellent performance of the operations and spacecraft, and to cope with the expected long duration of the mission, it has been decided to migrate the mission control system in both operations centers from the old, centralized Spacecraft Control Operating System I (SCOS-I), based on the virtual memory system (VMS) operating system, to the distributed, Unix-based, SCOS-2000 system [1].

The migration project was performed in parallel with live operations using the old control system. This was an extra overhead in the actual workload of the team,

*Data Systems Manager, ESA-ESAC, Madrid, Spain.
†Instrument Operations Team Leader, ESA-ESAC, Madrid, Spain.
‡Data Systems Manager, ESA-ESOC, Darmstadt, Germany.
§Spacecraft Operations Manager, ESA-ESOC, Darmstadt, Germany.

but it proved to be an excellent scenario to validate the migrated system. Given the distributed nature of the control system architecture, good coordination and maximum synergies were essential aspects of the project. The complex upgrade project began in late 2002 at ESOC and early 2003 at ESAC. Development work ran through late 2004. The project team involved flight operations and software engineers at ESOC and ESAC, as well as extensive support from industry teams.

Despite these challenges, the upgrade proceeded as planned, and by March 2005 a series of live tests had been completed, using the newly implemented SCOS-2000 software to control XMM-Newton and receive science data, while the older SCOS-I system was kept in operation. In June 2005, the final switchover was made, and the XMM-Newton ground segment project ended with a successful upgrade. The ultimate measure, however, was not only the technical success of the project but also the financial success.

This chapter describes how the overall objectives of the migration process have been achieved, including the preservation of the science quality, operational efficiency, system performance of both real-time and reprocessing activities, and external interfaces. The chapter also focuses on the validation strategy and test approach that were driven by the needs of minimizing the impact on science operations, avoiding any disruption to the science pipeline process and scientific data distribution cycle. Other migration aspects, external but related to the porting of the control system, such as adaptation of the flight operational procedures and auxiliary tools, are also analyzed.

A summary of the XMM-Newton observatory is presented in Sec. II, while the ground segment is outlined in Sec. III. The control system and associated dataflow are described in Sec. IV. The migration project is described in Sec. V. Detailed description of the extended validation is included in Sec. VI. Section VII summarizes the lessons learned. Finally, the conclusions are presented in Sec. VIII.

II. Observatory

XMM-Newton [2] is a high throughput X-ray observatory that is the second cornerstone of the ESA long-term scientific plan. With a total length of 10 m, the 4-ton spacecraft is the largest scientific satellite ever launched by the European Space Agency. The spacecraft was launched on 10 December 1999 on flight V119, the first commercial Ariane-5, with insertion into a highly eccentric, high inclination orbit, having a perigee height of 7000 km and an apogee height of 114,000 km. The operational orbit has an inclination of 40 deg and a period of 47.8 h. XMM-Newton has been operating as an open observatory, providing scientific data through three imaging cameras and two spectrometers, as well as visible and UV images through an optical telescope.

The X-ray telescope [3] is composed of three barrel-shaped mirror modules, each with 58 Wolter I mirrors, nested in a coaxial confocal configuration. This design provides a large collecting area equivalent to 4300 cm^2 at 1.5 keV, covering the spectral range of 0.1–12 keV through grazing incidence reflection.

XMM-Newton carries three X-ray imaging charge coupled device (CCD) cameras: the European Photon Imaging Cameras, each of them in the focal plane of one of the X-ray telescopes [4, 5]. The cameras are of two different types, one of them using a new type of CCD (pn) especially developed for X-ray missions.

Because all three cameras work in single-photon register mode and can register also energy and arrival time of each incoming photon, they provide simultaneously moderate spectroscopic and timing capabilities. Different operating modes can be used to optimize the observations according to the brightness of the target and the purpose of the observation.

Behind two of the three X-ray telescopes, diffraction gratings intercept around 50% of the incoming light, dispersing by reflection onto the CCD detectors of the reflection grating spectrometers [6], with spectral resolution $\lambda/\Delta\lambda$ ~200 in the first-order dispersion of the soft X-ray domain (0.3–2.4 keV).

Coaligned with the X-ray telescopes, the optical monitor [7] gives XMM-Newton a multi-wavelength capability, operating in the range 160–660 nm. The optical monitor camera can work also in photon counting mode, providing time-resolved information in addition to visible and UV images. The instrument presents a field of view of 17 arc min and a limiting sensitivity of 20.7 magnitude for an integration time of 1000 s. Additional optical and UV grisms provide XMM-Newton with a moderate spectral resolution in this wavelength range.

III. Ground Segment

The XMM-Newton ground segment is implemented in a distributed architecture, as shown in Fig. 1. Main components include the ground stations, required to provide a full coverage of the scientifically useful part of the orbit, the MOC, located at ESOC in Darmstadt, Germany, the SOC, located at ESAC in Villafranca del Castillo, Spain, and the Science Survey Centre (SSC), operated by a consortium led by the University of Leicester, England, United Kingdom.

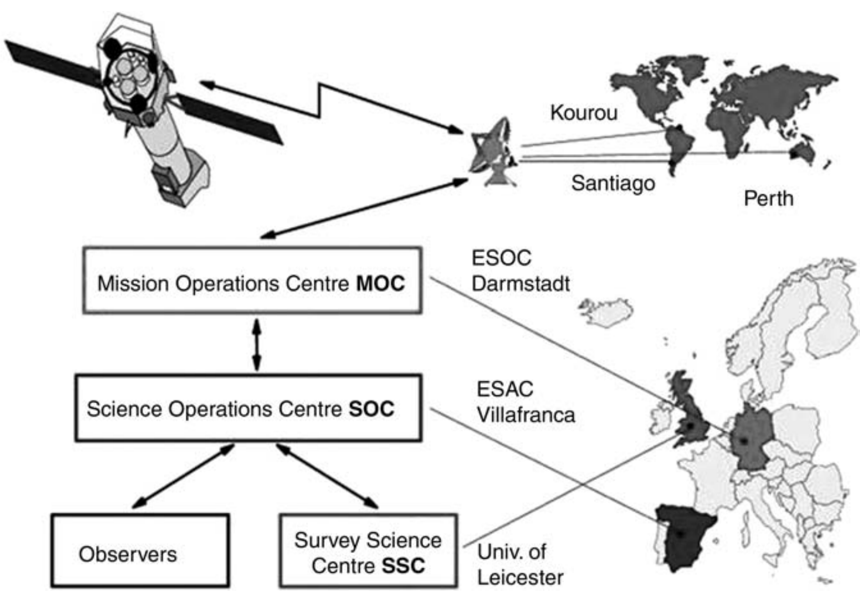


Fig. 1 XMM-Newton ground segment.

A. Ground Stations

The ground stations track the satellite and communicate at S-band with the onboard transponders. Three ground stations are used: Perth in Australia, Kourou in French Guiana, and Santiago in Chile. These stations were selected to ensure permanent communication with the satellite during the scientifically useful part of its orbit.

B. Mission Operations Center

The MOC is responsible for monitoring and controlling the satellite. All telecommands are sent from the MOC and all telemetry is received at the MOC. The main elements of the MOC are the following:

1) The mission control system, in charge of receiving, decoding, and processing the telemetry, as well as assembling the telecommands to the satellite. The control system is implemented in two separate chains, prime and backup, to provide the necessary redundancy of the operational elements.

2) The integration and validation chain that includes the XMM-Newton simulator, a software model of the satellite platform, used to validate commands and procedures.

3) The flight dynamics system in charge of orbit determination and attitude reconstruction.

C. Science Operations Center

The SOC is responsible for the science outcome, including the preparation of all scientific observations and analyzing and processing all scientific data. Main elements of the SOC are the following:

1) The science control system, in charge of planning the scientific observations and monitoring their execution. The science control system is implemented in two operational chains consisting of the prime and redundant systems.

2) In addition, a bulk-reprocessing capability is implemented in the redundant system to allow for the reprocessing of science telemetry using offline telemetry files.

3) The integration and validation chain that includes the XMM-Newton science simulator, a software model of the instruments, used to validate the science procedures as well as the instrument onboard software.

4) The archive management system, which stores and allows retrieval of data and associated procedures.

D. Science Survey Center

The SSC is responsible for the pipeline processing of the observations as well as performing serendipitous sky surveys and the compilation of the serendipitous source catalog.

IV. Control System

The XMM-Newton control system [8] is the kernel of the ground segment. Figure 2 is a schematic diagram of the complete dataflow, illustrating the main

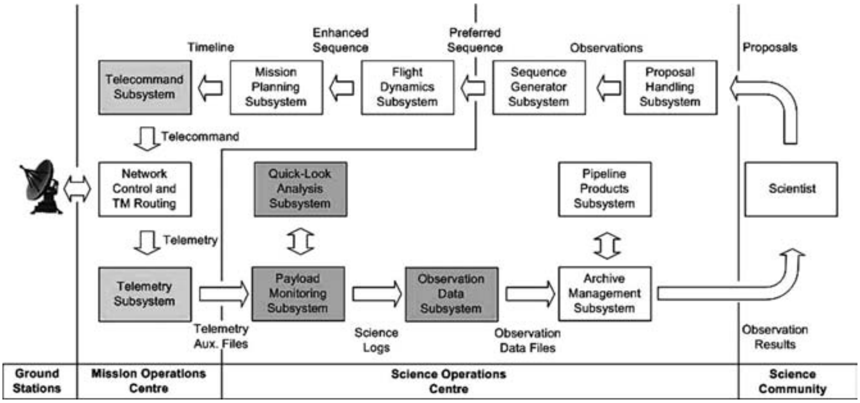


Fig. 2 XMM-Newton control system and associated dataflow.

components of the control system and related elements. Within the distributed architecture of the project, functional elements are distributed between MOC and SOC, as indicated in the figure.

A. Uplink Subsystems

The uplink part of the dataflow comprises the following subsystems:

1) Proposal handling subsystem: Collects the proposals for observation from external scientists and consolidates approved proposals into detailed observations.

2) Sequence generator subsystem: Produces command sequences required to perform the detailed observations received from the proposal handling subsystem. The resulting sequence of observations—preferred observation sequence in the figure—is then transferred to flight dynamics for further validation and expansion.

3) Flight dynamics subsystem: Validates and expands the preferred observation sequences, generating the enhanced preferred observation sequences. This subsystem is in charge of orbit determination and attitude reconstruction, and also defines optimal attitude and orbit maneuvers, respecting in-orbit constraints.

4) Mission planning subsystem: It performs the final processing and validation of the mission planning files, prior to their conversion into telecommand packets by the telecommand subsystem.

5) Telecommand subsystem: This subsystem allows the configuration of the telecommand chain, the preparation of telecommands for manual submission, the control of the prepared timeline of telecommands for automatic submission, the interface with the ground station equipment, the validation and verification of telecommands throughout the telecommand chain. It holds the history archive of the telecommands for the entire mission.

6) Network control and telemetry routing subsystem: It is a general purpose interface to the ground station network. It establishes the communications links between the control system and the ground station, enables telemetry and telecommand data transfers, as well as tracking data and ground station files, such as

orbit predictions and schedules, and monitors the status of the communication links.

B. Downlink Subsystems

The downlink components, indicated in the upper part of the figure, include the following:

1) Telemetry subsystem: It allows the processing, monitoring, and archiving of the incoming housekeeping telemetry, the correlation of the onboard computer time with coordinated universal time (UTC), the maintenance of an image of the onboard time-tag buffer, the display of the contents of any packet, and the details of all events and anomalies received.

2) Payload monitoring subsystem: This subsystem is responsible for the reception and processing of the telemetry received from the MOC, in both real-time mode and offline, in the reprocessing mode, via raw telemetry files. The main function is to extract and process science data from the incoming telemetry, and to store it into intermediate files—the science logs—in Flexible Image Transport System (FITS) format [9]. This subsystem is largely an instantiation of the telemetry subsystem and as such utilizes the SCOS-based infrastructure. Certain functionality is not available on the payload monitoring subsystem, while in other areas enhanced functionality is needed, such as processing of housekeeping telemetry.

Additional capabilities have been introduced in this subsystem to allow the processing of raw telemetry from data files on disk, incorporated into the bulk-reprocessing system.

The main tasks in the payload monitoring subsystem are the instrument processors, responsible for the processing of the science data from the instruments, including 1) generation of the science logs, and 2) generation of spontaneous parameters, comprising data extracted from the fixed part of the science telemetry, and statistical data calculated from the science data. The resulting output is usable by both the observation data subsystem and the quick look analysis.

3) Observation data subsystem: This subsystem generates raw science observations for each instrument, based on the output files produced by the payload monitoring subsystem. It comprises the observation data files and slew data files, as a set of files in FITS format [9] including 1) science logs, 2) housekeeping data files generated from telemetry stored in the history files, 3) spacecraft attitude and orbital data extracted from the flight dynamics subsystem, 4) time correlation information, and 5) observation summary information.

4) Quick look analysis: It is an auxiliary display subsystem to monitor the observation process. This subsystem allows real-time verification of the results of the observation being carried out, using the intermediate files produced by the payload monitoring subsystem. In addition, it allows selection of new satellite pointing information from the current observation.

5) Archive management subsystem: It stores raw science files generated by the observation data subsystem for further distribution. It is the central subsystem of the SOC that stores observations and slew data files. These files are then transferred to the SSC for science processing.

6) Pipeline products subsystem: Located at the SSC in Leicester, this subsystem generates the science products that will be transferred to the observer at the end of the downlink dataflow. The subsystem uses the science analysis subsystem to calibrate and produce scientific estimations of the observed events.

C. Common Subsystems

In addition to the uplink and downlink subsystems just described, there are common subsystems not indicated explicitly in the figure. These subsystems are the following:

1) Database subsystem: It manages the import, editing, generation, and export of the satellite control database, based on the satellite manufacturers database, updated with the flight operations procedures.

2) Onboard software subsystem: It maintains and controls the onboard software by importing, exporting, and comparing telemetry and memory images. It performs check sums on memory images and prepares and up-links telecommand images and displays and prints files stored in the system.

3) File transfer subsystem: It provides a means of transferring data files between peer processors on different host computers.

4) Long-term archive: This archive stores all of the telemetry and telecommand data during the mission lifetime, providing analysis facilities to allow instrument performance evaluation.

V. Migration

Because of the expected long duration of the XMM-Newton mission, with operations approved until 2010 and possible extension to 2014, it was decided to start the migration of the control system of both MOC and SOC from the old SCOS-I-based system to SCOS-2000 [1]. The main reason for the migration was to achieve a significant reduction in the maintenance costs of the control system. The SCOS-I infrastructure is dependent on the VMS operating system, implemented in the Alpha platform; it is difficult to maintain and with uncertain long-term support from the supplier.

In addition, the migration project aligns the control system with the strategic developments of ESA, as part of the mission families established for SCOS-2000. Thus, further cost reductions are expected from a common maintenance approach across missions within the same family.

Finally, the migrated system is implemented in modern equipment that provides improved performance and allows more flexibility in operations.

A. Migrated Subsystems

The subsystems to be migrated are highlighted in Fig. 2. They comprise all of those subsystems that use the SCOS-I infrastructure, including the telecommand and telemetry subsystems at the MOC, the payload monitoring, observation data subsystem and quick-look analysis at the SOC. In addition, the database subsystem

and the long-term telemetry archive, critical common elements of the mission control system, heavily dependent on the SCOS-I infrastructure, were also migrated.

Key aspects of the migration were to preserve science quality and operational efficiency of the mission while maintaining or improving system performance in real-time and reprocessing activities. Therefore, an essential requirement was to maintain system interfaces, as defined by the interface control documents. This backward compatibility of the migrated system ensures that only minimal changes are required to existing operational procedures. In addition, a proper implementation of the migrated database subsystem ensures that all telemetry reports and out-of-limit displays remain unchanged, and the same alarms, warnings, and information messages raised in the old system are also raised in the migrated system.

Another fundamental requirement was to minimize the amount of code to be migrated, by making use of basic functionality available in SCOS-2000. These functions include the generic telemetry and telecommand subsystems modified for XMM-Newton, and the generic onboard software management subsystem, also modified for XMM-Newton.

With this approach, only mission-specific components were ported to the new SCOS-2000 infrastructure. Critical elements were the replacement of SCOS-I interfaces by SCOS-2000 where possible and the automatic code conversion for instrument processors.

In the original control system, the mission database is implemented in Oracle, in two different instances, one at the MOC the other at the SOC. A complex merge procedure was used to update the contents of the SOC database with any modified contents of the MOC database. In the migrated system, the database subsystem has been streamlined by replacing this double system by one single master database implemented according to the design originally developed for the CryoSat project.

The long-term archive of the original control system is based on the spacecraft evaluation system, a telemetry archive implemented in VMS. This subsystem is replaced by the new telemetry data retrieval subsystem, implemented in Unix.

An essential aspect of the migration was to use the synergies between the two projects at MOC and SOC. The two migrations present many interdependencies, with large common subsystems that could be reused or adapted to the specific MOC or SOC environments. The main dependencies identified included the database subsystem, the long-term archive, the sharing of functions between telemetry and payload monitoring subsystems, handling of derived parameters and mimics displays. Substantial savings were achieved by appropriate code reuse in the migration of these subsystems.

Finally, another essential aspect of the migration was to adapt auxiliary tools to the new migrated system. Although the new system was backwards compatible with the original control system and interface specifications were strictly maintained, some of the offline subsystems, like the long-term archive, have been replaced by a new implementation. Therefore, new trend analysis tools had to be implemented [10], and modifications to other auxiliary tools such as the flight operations procedures and timeline tool, were required.

The SCOS-2000 baseline selected for the migration was Release 3.1, to align the control system with the most modern version, used by other similar missions at ESA.

B. Project Phases

The migration project was conceived as two different project instantiations, with different schedules, adapted to the specific needs and constraints of MOC and SOC. The migration of the MOC started in October 2002, while the SOC migration started in March 2003. The delay in the schedule was intentionally introduced, so that experience in MOC migration could be retrofitted in the SOC, and maximum synergies could be achieved by reusing and adapting the migrated code. Although in the original conception of the project the two control systems could migrate independently, it was agreed at an early stage that both, MOC and SOC, would switch together to the migrated systems. This decision allowed a more comprehensive validation of the migrated system, so that operational tests could be implemented at both places in a coordinated form.

The migration project was done in a phased approach, with incremental releases being produced, each offering additional functionality with respect to the previous release. The coordination between MOC and SOC was essential, so that despite the shift in milestones during the initial phases, synchronization was introduced at the latest phases of the project, allowing a proper coordination during the validation of critical functions.

Project phases for the two migration projects include the following:

1) Consolidation phase, to perform initial project analysis and update of technical notes, software requirements, and architectural design documents.

2) Incremental releases of the migrated subsystems. Initially, a total of four releases were identified, each with specific added functionality. The releases were associated to an acceptance test procedure to validate new functionality.

3) Parallel operations, a critical step in the validation process. During this phase, the migrated system runs in parallel with the SCOS-I-based control system.

4) Live operations, after successful validation of the migrated system.

In practice, three additional deliveries were scheduled at the MOC after July 2004. Following the successful validation of the system during the parallel operations phase, maintenance releases are delivered in combined, MOC-SOC releases, so that software validation and deployment can be synchronized at both sites, with minimal overheads.

VI. Validation

The validation of the migrated system had to demonstrate completeness and usability of the control system for the purpose of operating the XMM-Newton spacecraft. The end goal of the testing and validation process was to certify the migrated system as the fully functional control system for both the MOC and the SOC.

The migration project was carried out with major emphasis on the preservation of the science quality, operational efficiency, external interfaces, and system performances for both real-time and reprocessing activities. On this basis, the validation strategy and test approach were driven by the needs of minimizing the impacts on science operations and avoiding any disruption to the science pipeline process and scientific data distribution cycle. This implied that validation of the SCOS-2000 system was run in parallel with day-to-day spacecraft operations,

with minimum consumption of mission resources (e.g., minimum time subtracted to science) but with considerable overhead on the control team activities at both MOC and SOC. Given the distributed implementation of the control system, good coordination and maximum synergies between all parties involved, mainly software engineers, flight control teams at MOC and SOC, and flight dynamics engineers, during the migration process and especially in the acceptance and validation exercise, were essential.

Because the primary objective of the validation was to confirm that the system as a whole was suitable for operations, the test procedures were aimed at realistic operational scenarios, with a view to being essentially a regression test against the SCOS-I system. For this reason the procedures were not necessarily targeted to confirm individual software requirements, but to demonstrate that all operational functions were available, and could still be successfully exercised. Such an approach was also based on the fact that tests against requirements were performed by the development team, prior to software delivery.

The entire validation spanned the period from July 2003 to June 2005, divided into three main phases:

1) Development phase, in which MOC and SOC were validated mainly as isolated entities.

2) System test phase, where MOC and SOC were exercised together in complex and fully realistic test scenarios, specifically designed to validate the complete data flow and relevant system interfaces, the real-time science telemetry processing at the SOC, and satellite commanding from the MOC.

3) Parallel operations phase as the final stage of the validation process, where the SCOS-2000 system was operated in parallel with the SCOS-I system during a few months.

A. Development Phase

During the development phase, the system functions were migrated incrementally up to the delivery of a complete control system. Several bug-fixing releases and/or individual patches were subsequently required to correct for encountered problems and deficiencies. This phase lasted from mid-April to the end of November 2004 at the MOC, and from mid-September to the end of November 2004 at the SOC. The primary objective was to perform installation, testing, debugging, and acceptance of the control system. During this phase, the control teams at MOC and SOC analyzed the limitations and advantages of migrating the control system. In addition, it was a critical period to build up the relationship with software engineers during testing and troubleshooting.

At the MOC, testing proceeded with various test setups of increasing complexity, such as stand-alone configuration with SCOS-2000 server and client(s) decoupled from the operational environment, passive parallel configuration with telemetry fed from the active network control and telemetry routing subsystem into both SCOS-I and SCOS-2000 systems, with SCOS-I as the prime system and SCOS-2000 operated in listening mode, as the shown in Fig. 3, while for exercising the commanding function the simulator was extensively used. Timeline execution tests were used for wider regression of each release. The test procedures were grouped into the following four types:

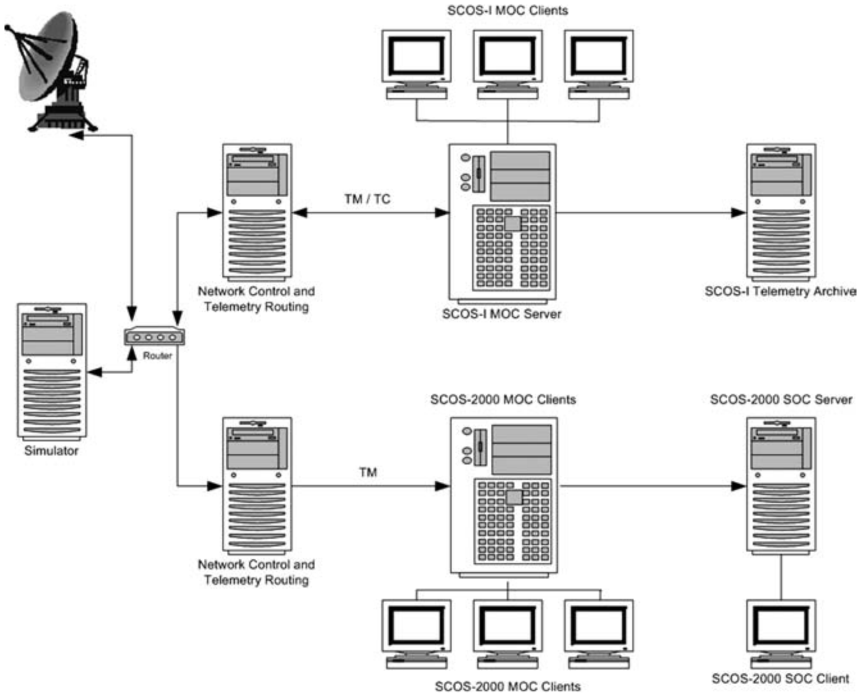


Fig. 3 Configuration during the development phase at the MOC.

1) Configuration procedures to confirm that the full functionality of SCOS-I was available in the SCOS-2000 environment prior to detailed verification.

2) Performance procedures for extensive testing of all system tasks. In particular, testing of the telecommanding function was performed using the simulator with predefined test scenarios. In general, whenever possible, the SCOS-2000 system at the MOC was run in passive parallel configuration. The operators were responsible for confirming that all alarms received on SCOS-I had a counterpart on SCOS-2000.

3) Operational procedures to validate routine operational aspects. The tests were designed to exercise as much as possible all operational scenarios that the XMM-Newton operations may face, before introducing the migrated systems into full real-time control of the mission, such as target of opportunity, reaction to radiation alerts, early termination of observations, eclipse operations, recovery from satellite safe mode, observation of bright objects, etc.

4) Interface control procedures to confirm that the system interfaces were consistent with the SCOS-I interfaces. It was essential to confirm that all flight dynamics system products could be generated and transferred using the migrated facilities. The ability to interchange information and to transfer the different files and products had to be thoroughly demonstrated. This included telemetry acquisition by flight dynamics, attitude recovery, star field generation, orbit maintenance, preplanned skeleton file generation and transfer, preferred observation sequences

reception, enhanced preferred observation sequences generation, timeline generation, slew generation, generation of attitude history files, etc.

At the SOC, the main objective of the acceptance tests was to verify the correct processing of science telemetry and proper generation of scientific products by instrument processors, although other areas of functionality, such as long-term archive and the integration of the quick-look analysis with SCOS-2000 were extensively validated and exercised, since these are essential components for supporting routine science operations and for the online and offline monitoring of the instrument status. Validation of science products was based on the comparison of the outputs of the original and migrated systems using identical raw telemetry files as input. This could best and most efficiently be achieved through reprocessing of selected raw telemetry datasets with both the original and the migrated systems, and the systematic comparison of results at each stage of the process. Twenty-four past observations, covering all instrument modes and science telemetry flavors, were used in the first instance to accomplish this task. To guarantee maximum coherence of the datasets, all 24 observations used for direct comparison were reprocessed in both systems in an identical environment, i.e., the same operational database, access to the same archive management subsystem, and the same input files from SCOS-I raw telemetry files. Because of the limitations of the XMM-Newton simulator in generating meaningful scientific telemetry, the validation of the science products produced in real-time was postponed to the system validation phase. It was assumed that because of the commonality of the real-time and reprocessing functions, most of the problems detected, filtered and corrected at the reprocessing level, were all applicable to the real-time level, too. This allowed the validation team to anticipate and correct problems in advance to actual testing.

The database delivered with SCOS-2000, common to both MOC and SOC, was an import of the SCOS-I database suitably reorganized for the migrated environment. This subsystem required extensive testing, addressing mainly the following areas:

1) Inspection and comparison of the converted database with the SCOS-I database, to confirm completeness, integrity, correctness and consistency.

2) Testing of the new database editors in typical operational scenarios.

3) Testing of interfaces, to prove feasibility of remote maintenance across the MOC-SOC link and to confirm interoperability with the SOC mission planning system through the archive management subsystem.

4) Performance testing consisting of implicit usage of the database contents when operating in the SCOS-2000 environment.

B. System Validation Phase

The system validation phase lasted from the end of November 2004 to the beginning of March 2005. The primary objective was to validate the following major system components and operational aspects: 1) MOC-SOC interface through connection via the MOC/SOC link between ESOC and ESAC; 2) production of raw telemetry files by the MOC and their usability at the SOC for reprocessing; 3) real-time processing of science data; 4) end-to-end MOC-SOC-SSC operational cycle dataflow; and 5) commanding of all instrument modes to confirm correctness of the operational database at both MOC and SOC and its full consistency with the mission planning system.

To reach these objectives, a certain number of test windows around perigee were used, without impacting on science activities. In addition, a few revolutions were devoted entirely to testing. These were the so-called All Mode Test revolutions. The plan originally called for three revolutions in total, eventually extended to five, as follows:

1) One All Mode Test revolution: Rev-909 on 24–26 November 2004, to exercise all scientific instruments modes; SCOS-I was used for commanding with parallel telemetry and parallel passive execution of timeline file on SCOS-2000 system. This test proved that all types of possible science telemetry could be successfully processed in real time by the control system at the SOC; in fact, more than 700 FITS files were properly produced.

2) An additional All Mode Test revolution: Rev-940 on 25–27 January 2005, designed to have the same operational layout as for the previous case but with commanding from the migrated system. It was proven that all instrument command blocks currently stored in the mission planning system for the execution of all defined instrument modes (scientific, engineering, special modes used for health checking and/or calibration) could be successfully executed. In addition, manual spacecraft commanding was successfully exercised, too, e.g., timeline rejoining operations, optical monitor recovery from double bit memory error condition. Comparison of SCOS-I and SCOS-2000 real-time functionality, such as coherence of out-of-limits and onboard report messages and functioning of critical derived parameters for radiation monitoring, showed no significant discrepancies between the two systems. However, because of instability of the MOC-SOC link, that was subsequently corrected, and because of other non-SCOS-2000 related problems, the test was repeated in two further occasions including Rev-954 on 22–24 February 2005 and Rev-958 on 2–4 March 2005.

3) One revolution at the end of the winter 2005 eclipse season: Rev-950 on 14–16 February 2005, to exercise eclipse operations and calibrations of the attitude and orbit control subsystem.

A remarkable output of this phase, since we were using real operational scenarios, was the increased level of confidence the MOC and SOC control teams had with the migrated system, and perhaps more important, the confidence that solutions could be found to future problems in parallel operations phase.

C. Parallel Operations Phase

The parallel operations phase lasted from March 2005, as of Rev-961 inclusive, to 14 June 2005, when the migrated system was declared operational and used thereafter for controlling the XMM-Newton spacecraft. Parallel operations coincided with finalization of operator training and certification, final installation and deployment of all associated hardware, and completion of procedure updates. During the parallel operations period, the SCOS-2000-based system at both MOC and SOC was used in parallel to the SCOS-I system (upper part of Fig. 4) during an uninterrupted period of time to confirm its stability and reliability, to validate final fixes, to confirm full coherence between SCOS-I and SCOS-2000 science output through systematic comparison of SCOS-2000 and SCOS-I real-time science products, to confirm repeatability of data reprocessing through systematic comparison of SCOS-2000 real-time and reprocessing of science products, and finally

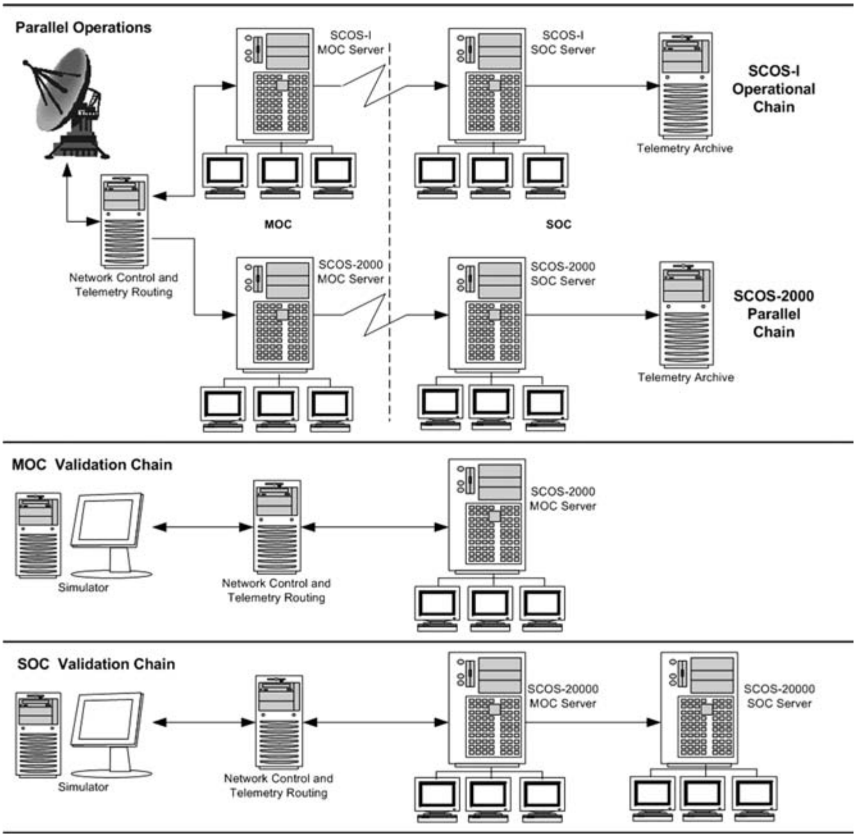


Fig. 4 Configuration of MOC and SOC during the parallel operations phase.

to assess the SCOS-2000 system performances in comparison with SCOS-I. It is worth emphasizing that the tests during this phase did not impact the overall science efficiency of the mission.

The migrated system is operational at both MOC and SOC as of 14 June 2005. Since then, the global functionality and performance of the system has been fully nominal, confirming the excellent level of extensive validation and tuning at both MOC and SOC.

VII. Lessons Learned

A migration project is a very special enterprise: the flight control teams and the spacecraft and instrument controllers are adapted to the old system, they know how to live with its limitations, and they have experience with operational work-arounds when needed. In addition, a number of tools are available to implement operational features as required by the old system. Therefore, in a migration project it is essential to identify the advantages of the new system, to compensate for the effort required from implementers and end users in the migration process.

However, if the system works and performs according to specifications, it is not perceived as a merit, because it was already working.

The migration of the XMM-Newton control system was a successful project because it was delivered on time, below budget, and with satisfied customers. Key points for the success of the project were the following:

1) Management supported the project, i.e., it was planned in advance, with adequate funding.

2) All project actors were motivated and deeply involved in the migration process to adequately use their experience.

3) Especially remarkable was the cooperation and motivation of the flight control teams at MOC and SOC.

4) A very important point was the positive relationship with the software engineers, allowing a smooth collaboration. The collaboration was understood as 1) problem sharing and 2) involvement in the decision process.

5) An essential aspect for the success of the project was the technical coordination of the combined software development process at MOC and SOC.

6) The quality of the system was ensured by extensive testing and product verification. This is an interesting feature in a migration project. Realistic test cases and operational scenarios can be executed in parallel with the old and migrated systems.

7) An essential factor for the quality of the software development process was a strict configuration control, with consistent software reviews shared between MOC, SOC, and external developers, and the proper distribution of information, with a dedicated project management system.

8) Finally, the project coordination within this distributed architecture was ensured by common software releases during the latest phases of the project, and agreed timelines for software validation and deployment.

In the continuous evolution of the technology, it is unavoidable that projects with a foreseen long duration will need to migrate to new infrastructures, to optimize maintenance and to reduce operations costs.

As a last remark, once one decides such an enterprise, it is important to maintain the scope of the migration project, avoiding additional ad hoc migrations and upgrades, triggered by the success of the main migration project. These additional migrations may endanger the original project and should be avoided by all means, probably scheduling them at a later stage.

VIII. Conclusion

The challenge of the project was that no one had ever replaced the mission control software in mid-mission before, and senior management had requested that the upgrade would not interrupt the flow of valuable science data from XMM-Newton.

The new migrated system has been in operations since 14 June 2005. The new mission control system, based on SCOS-2000, was introduced without interrupting the critical mission dataflow; output science files were continuously transferred to the SSC for pipeline processing at the University of Leicester.

Project objectives have been achieved on time and within budget; in fact, current estimates indicate 25% savings. The project also achieved a full customer satisfaction, a very difficult result because of the complex nature of the user community that includes the flight control teams at MOC and SOC, the scientists at SOC and SSC, and ultimately, the science community worldwide.

Project requirements, schedules, and budgets are easy to track and to measure. The difficulty in a project resides in quantifying customer satisfaction. As a final indicator of this difficult estimator, we quote an anonymous manager that, after completion of the parallel operations phase, voluntarily declared: “Our operators at ESAC are very happy about the porting and they would not like to go back to SCOS-I!”

Acknowledgments

The project has been successful because of the excellent work of the combined migration team. In addition to the authors of this contribution, the following persons contributed to the success of the project: the flight control team, including R. Muñoz, J. Fauste, S. Rives, B. Olabarri, G. Buenadicha, and N. Cheek at the SOC, and O. Ojanguren, J. Martin, A. Guidi, S. Anstoetz, and R. Pérez Bonilla at the MOC; the software engineers, including J. Hoar, M. Couch, P. Tharagonnet, N. Burdett, R. Norbury, M. Loveday, and J.C. Vallejo. Spacecraft and instrument controller teams at MOC and SOC were essential actors during validation and parallel operations phases. Finally, the authors acknowledge the continuous support from management.

References

[1] Kaufeler, J.-F., Jones, M., and Karl, H.-U., “Promoting ESA Software as a European Product: The SCOS-2000 Example,” *ESA Bulletin 108*, Nov. 2001, pp. 72–77.

[2] Jansen, F., Lumb, D., Altieri, B., Clavel, J., Ehle, M,. Erd, C., Gabriel, C., Guainazzi, M., Gondoin, P., Much, R., Munoz, R., Santos, M., Schartel, N., Texier, D., and Vacanti, G., “The XMM-Newton Observatory – I. The Spacecraft and Operations,” *Astronomy and Astrophysics*, Vol. 365, No. L1, Jan. 2001, pp. L1–L6.

[3] Aschenbach, B., Citterio, O., Ellwood, J. M., Jensen, P., de Korte, P., Peacock, A., and Willingale, R., “The High-Throughput Spectroscopic Mission. Report to the Telescope Working Group,” ESA SP-1084, Feb. 1987, pp. 22–25.

[4] Strüder, L., et al., “The European Photon Imaging Camera on XMM-Newton: The pn-CCD Camera,” *Astronomy and Astrophysics*, Vol. 365, No. L18, Jan. 2001, pp. L18–L26.

[5] Turner, M. J. L., et al., “The European Photon Imaging Camera on XMM-Newton: The MOS Camera,” *Astronomy and Astrophysics*, Vol. 365, No. L27, Jan. 2001, pp. L27–L35.

[6] den Herder, J. W., et al., “The Reflection Grating Spectrometer On Board XMM-Newton,” *Astronomy and Astrophysics*, Vol. 365, No. L7, 2001, pp. L7–L17.

[7] Mason, K. O., et al., “The XMM-Newton Optical/UV Monitor Telescope,” *Astronomy and Astrophysics*, Vol. 365, No. L36, Jan. 2001, pp. L36–L44.

[8] Ashworth, C., Haddow, C. R., Pendrix, E., and Retbøll, L., “The XMM Mission Control System,” *The 5th International Symposium on Space Mission Operation and Ground Data Systems*, Tokyo, June 1998, http://track.sfo.jaxa.jp/spaceops98/ops/98papers_e.html.

[9] “Definition of the Flexible Image Transport System (FITS),” NASA Science Office of Standards and Technology, NOST 100-2.0, March 1999.

[10] Fauste, J., Barreto, J., Casale, M., and Ponz, D., “An Interactive Telemetry Data Analysis Tool for XMM-Newton,” *SpaceOps 2006 Conference*, Rome, Italy, 2006.

IV. Communications and Tracking

This page intentionally left blank

Chapter 15

Mars Reconnaissance Orbiter Ka-Band Demonstration: Cruise Phase Operations

Shervin Shambayati,* David Morabito,† James S. Border,‡
and Faramaz Davarian§

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

Nomenclature

E_s = received energy per symbol, J
P_c = received carrier power, W
N_0 = single side noise spectral density, W/Hz

I. Introduction

NASA's Mars Reconnaissance Orbiter (MRO) is carrying a full suite of 32 GHz (Ka-band) telecommunications equipment to demonstrate the feasibility of Ka-band use for deep space science data return. This demonstration is necessary because the 50 MHz of bandwidth allocated at 8.41 GHz (X-band) is too small to handle the higher data rates expected from future deep space missions. (The frequency allocation at 32 GHz Ka-band is 500 MHz). The Ka-band link is more susceptible to severe weather events. Therefore, the operations concept that will be validated through this demonstration is based on maximizing the average data return on the Ka-band link subject to a minimum availability [1–3].

It was decided that during the cruise period various ground and spacecraft functions would be verified through 10 dedicated Ka-band demonstration passes. In addition to these, several delta differential one-way ranging (ΔDOR) passes as well as a number of "shadow" passes were scheduled. Shadow passes are passes

*Principal Investigator, Mars Reconnaissance Orbiter Ka-Band Demonstration.
†Senior Staff.
‡Principal Engineer.
§Manager, Deep Space Network Technology Program.

during which the X-band and the Ka-band links on the spacecraft were identically configured and were tracked by a ground antenna capable of receiving both X-band and Ka-band.

As a result of these passes, it was determined that MRO is fully capable of supporting the Ka-band demonstration activities during the two-year primary science phase (PSP). The Deep Space Network (DSN) also performed well despite some minor issues with the Ka-band monopulse active antenna pointing and with receiving and archiving of monitor data. These issues are expected to be resolved before the start of the PSP activities.

This chapter is organized in the following manner. In Sec. II an overview of the demonstration along with a brief description of the spacecraft and the ground system capabilities are given. In Sec. III, an overview of the Ka-band link telemetry and navigation performance during the 10 dedicated passes and the shadow passes is provided. In addition, performance of the ground system in terms of antenna pointing and measuring of the signal-to-noise ratio (SNR) is considered. In Sec. IV the issue of measuring the spacecraft equivalent isotropic radiated power (EIRP) is looked at in more detail. Section V covers the ΔDOR performance of the Ka-band during the cruise. Finally, in Sec. VI conclusions are reached.

II. Demonstration Overview

A. Demonstration Objectives

The objectives of this demonstration are to validate the proposed Ka-band operations concept for deep space missions and to modify this operations concept according to experience. Furthermore, this demonstration is to identify possible shortcomings in theground systems for tracking of Ka-band and propose remedies for them [4, 5].

The objective of the passes assigned to the Ka-band demonstration during cruise is to verify that both the spacecraft and the ground systems have the necessary functionalities for the Ka-band demonstration activities during the PSP. In addition, the cruise passes will familiarize the Ka-band demonstration team with project procedures and interfaces. This will allow the Ka-band activities to be executed smoothly during the PSP.

B. MRO Spacecraft

The Mars Reconnaissance Orbiter was launched from Kennedy Space Center on 12 August 2005 and went into Mars orbit on 10 March 2006. The spacecraft will finish its aerobraking maneuvers by September 2006, after which it will go through a series of calibration activities. From 7 October 2006 through 7 November 2006, the spacecraft will be in superior solar conjunction. During this time communications with the spacecraft will be limited and spacecraft operations will be kept to a minimum. From 8 November 2006 through 18 November 2008 the spacecraft will be in its PSP. During the PSP, the spacecraft will gather more data on Mars than all of the past missions to Mars combined. During the solar conjunction period, the Ka-band demonstration is allocated, on the average, one pass per day. During the PSP, the Ka-band demonstration is allocated two passes a week

and one ΔDOR pass a month [6]. In addition, the project will use Ka-band to transmit low priority science data for mission enhancement.

The Ka-band suite on MRO consists of a 35-W traveling wave tube amplifier (TWTA) and a 3-m parabolic antenna that produces an EIRP of 101.3 dBm, based on prelaunch measurements. By comparison, the X-band system has one primary 100-W TWTA and one backup 100-W TWTA with the same 3-m dish producing an EIRP of 96.2 dBm, based on prelaunch measurements. The reason that the Ka-band system has a larger EIRP is entirely due to the higher gain of the antenna at Ka-band [7]. The X-band signal could be transmitted on two low-gain antennas (LGAs) as well. There are two small deep space transponders (SDSTs) on the spacecraft (again, one as a backup) for modulation of the data. A simplified block diagram is shown in Fig. 1.

Although the spacecraft is capable of using both turbocoding with block length 8920 bits and rates 1/2, 1/3, and 1/6 and Reed-Solomon (RS) coding [concatenation with (7,1/2) convolutional code is done by the SDST], there are limitations on simultaneous X-band and Ka-band operations. These limitations are the following:

1) If the data transmitted over the Ka-band link are different from that transmitted over the X-band link, one link has to use turbocoding; the other has to use RS or concatenated coding.

2) If the data transmitted over the Ka-band link are different from that transmitted over the X-band link, the combined channel symbol rate of the two bands should be no greater than 6 Msps (megasymbols per second). For concatenated codes this includes the symbol rate increase due to the use of (7,1/2) code.

3) If the same data are transmitted over both the X-band and the Ka-band links, then the symbol rate on each channel cannot exceed 6 Msps. For concatenated codes this includes the symbol rate increase due to the use of (7,1/2) code.

4) The turbocoding option can support a maximum of 1.5 Mbps with rate 1/2 code due to ground decoder hardware limitations.

The ranging modulation and data modulation index for each band are independently configurable. For symbol rates above 2 Msps, quaternary phase shift keying (QPSK) modulation is used for X-band due to spectrum limitations. For Ka-band binary phase shift keying (BPSK) modulation is always used.

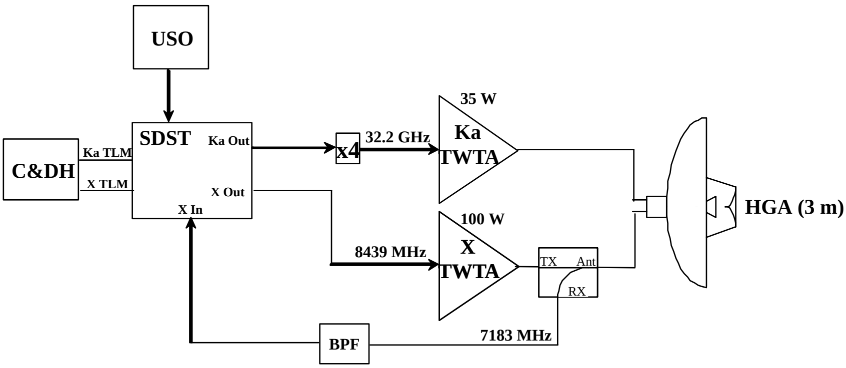


Fig. 1 MRO telecommunications and commanding subsystem.

DOR tones can be modulated on both X-band and Ka-band. Ka-band uses wider DOR tones than X-band because of availability of more spectrum, thus providing improved ΔDOR performance over X-band (see Sec. V).

The spacecraft is sequenced (programmed) primarily through two different procedures: background sequencing and mini-sequencing. Background sequencing programs the spacecraft for 28 days. Mini-sequencing programs the spacecraft for specific events such as calibrations of instruments and trajectory correction maneuvers. In addition, mini-sequencing is used to modify the background sequence according to the latest information available to the project. The background sequence usually takes 28 days to develop. Development times of mini-sequences vary depending on the nature of the sequence, but they take at least a week. Real-time commands could also be used with the spacecraft. However, their use is very limited as all spacecraft sequences must be validated before their actual implementation. All Ka-band activities during the cruise were sequenced as part of the background sequence with the exception of some real-time commands for modulation index changes. We expect to use background sequences for practically all Ka-band demonstration activities during the PSP.

The basic requirements on spacecraft for the Ka-band demonstration are the following:

1) The spacecraft must be capable of producing adequate EIRP for Ka-band (greater than 100 dBm).

2) The spacecraft's Ka-band link must be fully configurable in accordance with "MRO Telecom Design Control Document" [7].

3) The link's data rate could be changed during a pass according to the background sequence.

4) The link's modulation index could be changed during a pass according to the background sequence and real-time commands.

5) The spacecraft must be capable of producing wideband DOR tones for Ka-band.

6) The spacecraft must be capable of modulating uplinked ranging tones on the Ka-band downlink.

The data rate change requirement is a necessity for the Ka-band as the link performance changes significantly as a function of elevation. Through modulation index changes in real time, we will emulate the SNR changes that would result from real-time data rate changes. In addition, modulation index changes allow us to change the SNR. This will allow us to obtain SNR thresholds for different coding types.

C. Ground Systems

Part of the Ka-band demonstration is to assess the readiness of the DSN to track Ka-band signals from deep space missions. It should be noted that the DSN has tracked Ka-band only sporadically for Cassini radio science activities on a best-effort basis and that the MRO Ka-band demonstration will allow the DSN to track Ka-band telemetry regularly for the first time. Therefore, we expected that some challenges would arise with the DSN Ka-band tracking and that the passes during

the cruise will be used to identify problems that may exist with the DSN Ka-band equipment and operations.

The DSN consists of three Deep Space Communication Complexes (DSCCs) located at Goldstone, California; near Canberra, Australia; and near Madrid, Spain. These sites were selected by NASA to provide both 24-h coverage for missions that need them and to provide north-south coverage for the DSN.

There are four Deep Space Stations (DSSs, as the DSN antennas are called) in the DSN that are capable of receiving Ka-band. All of these antennas are part of the DSN 34-m beam waveguide (BWG) subnet. These are DSS-25 and DSS-26 at Goldstone, DSS-34 at Canberra, and DSS-55 at Madrid. During the Ka-band demonstration, we expect to use all of the stations with roughly equal numbers of passes divided among the three complexes (not the stations). The gain and noise temperature characteristics of these antennas are listed in [8].

There are two sets of capabilities that are required for the ground systems: those related to individual antenna performance and those related to the complex's signal and data processing capabilities. The basic antenna functions required for Ka-band demonstration are as follows:

1) The stations' Ka-band low-noise amplifiers (LNA) must meet the performance specifications in [8].

2) The blind pointing of the station for Ka-band must be better than 10 mdeg so that the active pointing (monopulse system) will be able to operate.

3) The monopulse must be operational at all of the stations for all of the passes.

It should be noted that the 34-m BWG Ka-band beamwidth is rather narrow (less than 18 mdeg). Therefore, the ground antenna pointing needs to be very good. Without the active antenna pointing, there could be 4 or 5 dB of loss in the link performance due to pointing errors.

The required signal and data processing functions are as follows:

1) DSCCs must be able to demodulate and decode Ka-band telemetry.

2) The ground receivers must accurately measure the system noise temperature (SNT).

3) The ground receivers must accurately measure SNR, especially symbol SNR (SSNR).

4) DSCCs must be able to measure Doppler and perform two-way ranging with the Ka-band signal.

5) DSCCs must be able to receive Ka-band DOR tones and perform ΔDOR measurements at Ka-band.

6) Monitor data from each pass must be delivered to the MRO query servers for each pass from the DSCCs with the proper sampling rate (once every 5 s).

It should be noted that for normal spacecraft operations only demodulation and decoding of the data, Doppler and ranging measurements, and ΔDOR measurements are required. The reason for the additional requirements is that the MRO Ka-band demonstration needs to identify those data outages that are caused by weather events and separate them from outages caused by other phenomena such as errors in the ground antenna pointing. For this purpose, the analysis will consist of correlating decoding errors with drops in the SNR and increases in the SNT. Therefore, accurate reporting of the SNR and the SNT,

as well as proper delivery and archiving of the monitor data, are needed for this demonstration.

III. Ka-Band Telemetry Passes During Cruise

In this section an exposition of Ka-band activities during those passes over which Ka-band telemetry was received is discussed. These passes fall into three categories: 1) dedicated Ka-band passes, 2) Ka-band shadow passes around trajectory correction maneuver 2 (TCM-2), and 3) Ka-band shadow passes around gravity science calibration 2. Each of these categories is treated separately.

A. Dedicated Ka-Band Passes

There were 10 passes dedicated to Ka-band demonstration during which both the ground system and spacecraft were put through their respective paces. While a pass-by-pass description is beyond the scope of this chapter, a general account of the performance of the spacecraft and the ground system is given here.

Of the 10 passes, DSS-25, DSS-34, and DSS-55 had three passes each, and DSS-26 had one. During these passes the spacecraft data rate, coding, and modulation index on Ka-band were changed during a pass using the background sequence. Also, the modulation index on Ka-band was changed using real-time commands. In addition, the Ka-band signal was turned off and on during one pass to simulate occultations around Mars, forcing the DSN to reacquire the Ka-band signal from the spacecraft repeatedly. Overall, the spacecraft performed flawlessly, and all of its required functions with the exception of the spacecraft Ka-band EIRP (see Sec. IV) were readily verified. On the Ka-band pass on day 05-304 (31 October 2005) over DSS-55, MRO set a planetary mission record for largest amount of data received in a day (116 Gbits) and also for the highest data rate ever (5.2 Mbps) from a planetary spacecraft using the Ka-band link. Also, on the Ka-band pass on day 05-280 (7 October 2005) over DSS-25, MRO became the first Jet Propulsion Laboratory (JPL) mission to transmit turbocoded data, again on Ka-band.

Ground systems functions, however, have not been fully validated. As mentioned before, this was not unexpected as the DSN had not tracked Ka-band on a regular basis and that one of the objectives of the cruise activities was to identify potential problems with the DSN so that they could be fixed before the start of the PSP.

One of the early problems that was encountered was the inaccurate reporting of the SSNR and carrier SNR (P_C/N_0) on both X-band and Ka-band due to high spacecraft received power and interference from the ranging tones. These errors in the SNR measurements along with high received signal power caused the SNT to be reported erroneously as well. Because the SNT measurements are based on a total in-band power measurement, to calculate the SNT, the SNT needs to be known very accurately in cases where the signal power is greater than or equal to the noise power in the band over which the SNT is estimated. As expected, as the spacecraft moved farther away from Earth, the problems with the SNT and the SNT reporting became less pronounced because of reduced received power.

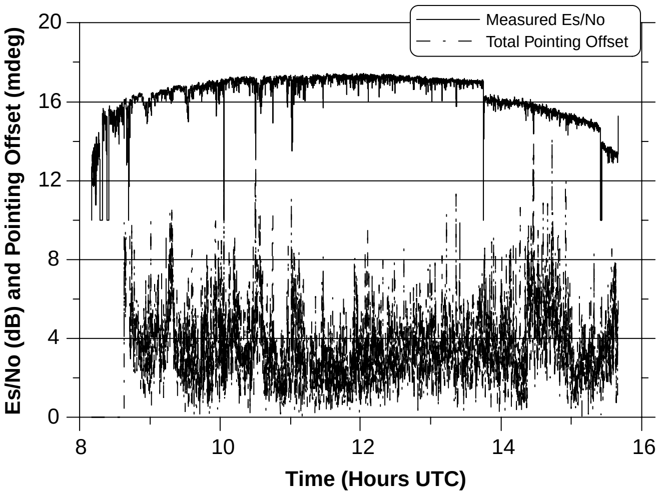


Fig. 2 Day 05-319 Ka-band E_s/N_0 and monopulse pointing offsets.

The monopulse active pointing system has functioned properly for all four antennas but not for all of the passes (monopulse was operational for 7 of the 10 dedicated Ka-band passes). Hardware problems at DSS-55 and lack of operational experience by the DSN personnel contributed to this. The hardware problems at DSS-55 have now been fixed. However, operational experience is necessary to validate this fix. In addition, a performance anomaly at DSS-34 on day 05-319 was observed. As Fig. 2 shows, there were drops in the measured SSNR (E_s/N_0) that correlated very closely with large monopulse pointing offsets. This was only the second time that DSS-34 was tracking the MRO Ka-band signal. Therefore, the monopulse equipment may not have been properly calibrated. Further analysis of this problem is required.

While the monopulse may not have worked perfectly at all times, based on the results obtained when monopulse was working, DSS-34 had the best blind pointing performance for the parts of the sky where MRO was tracked with pointing offsets of roughly 4 or 5 mdeg most of the time. This is attributed to a more rigorous blind-pointing calibration of DSS-34 before its commissioning for Ka-band. Other stations do not have as good a blind pointing as DSS-34; however, their performance was adequate for monopulse operations during the cruise.

There have also been problems with the monitor data being delivered fully to JPL and to MRO query servers. There were two passes at DSS-55 (day 05-325 and day 05-339) and one pass at DSS-34 (day 05-353) for which the monitor data were sampled rather slowly (several minutes between monitor updates compared to 5 s between updates under normal conditions, see Fig. 3). The reason for this slowness has not been determined yet.

The DSN has been notified of problems with the monopulse and the monitor data updates and is working toward solving them. Based on the progress that has been made up to now, it is expected that by the beginning of the PSP these issues will be fully resolved.

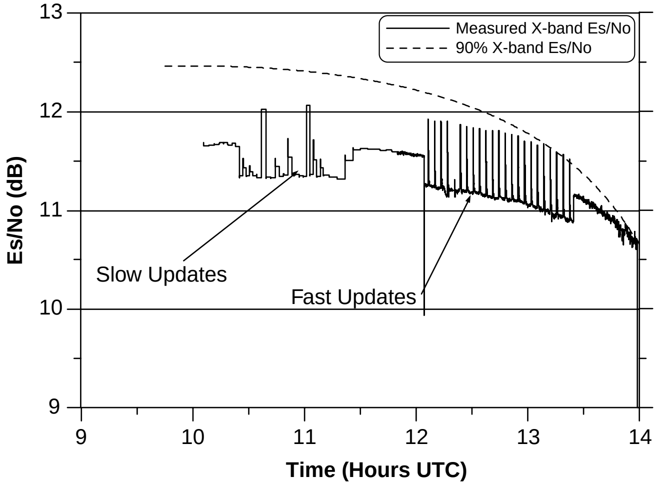


Fig. 3 Day 05-353 X-band E_s/N_0.

The DSN processed Ka-band telemetry as expected. However, it should be noted that the link was never stressed as it would be under nominal operating conditions during the PSP. In addition, a peculiar phenomenon was observed on a few of the passes. When the link was operating with 35-deg ranging modulation index and low data modulation indices (30–40 deg), periodic frame errors on the link were observed even though the link had more than 6 dB of margin. The periodicity of the errors corresponded to the ranging tone periodicity. This indicated interference from the ranging tones. Because of this, low modulation indices with high-ranging modulation indices will not be used on the Ka-band link during the PSP.

In addition to the telemetry demonstration, the MRO team also evaluated the performance of the Ka-band (and X-band) radiometric data types used for navigation and radio science during the cruise. These radiometric data types included Doppler and ranging. The downlink Doppler for both the X-band and the Ka-band (coherent with an X-band uplink signal) performed better than the specified navigation requirements. The scatters on the residuals of the two downlink Doppler signals were comparable between the bands, suggesting that the dominant error sources were either nondispersive or due to charged particle effects on the common X-band uplink signal. The performance on the X-band and Ka-band downlink ranging signals (both coherent with an X-band uplink) was also characterized and found to be comparable between the two bands, with the residual scatter and bias not exceeding the specified navigation requirements. In addition to the coherent data types, one-way Doppler performance using the ultra-stable oscillator (USO) and the auxiliary oscillator (AUX OSC) onboard the spacecraft was also characterized and found to be consistent with expectations based on specifications or preflight measurements [9]. In addition to analyzing performance of the individual X-band and Ka-band frequency bands, the difference between simultaneous X-band and Ka-band downlink frequency data was examined for the purpose of

identifying band-specific or dispersive error sources. Application of media calibration techniques on the data is currently being investigated for the purpose of realizing any improved performance.

Finally, these passes helped the Ka-band demonstration team and the spacecraft sequencing team to fully understand each other's modus operandi. For example, bit rate that the sequencing team uses is referenced to the input to the SDST. As the turbo-encoding of the data occurs before SDST, this means that the sequencing team specifies the data rate for turbocodes in terms of encoded symbol rate. The Ka-band team, however, had initially thought that the turbocode data rates were specified in terms of the information bit rate. During the cruise passes this issue was clarified.

B. TCM-2 Shadow Passes

For these passes, the Ka-band team used the opportunity that the spacecraft was being continuously tracked from 14 November through 20 November 2005 to have the spacecraft send down telemetry on the Ka-band link whenever the spacecraft was being tracked by a Ka-band capable antenna. The Ka-band configuration for these passes was identical to the X-band configuration, i.e., 550 Kbps concatenated coded data (1.1 Msps, 480 Kbps information data rate) with a 72-deg modulation index. The ranging was turned off for Ka-band but was turned on for X-band with a ranging modulation index of 17.5 deg. The monopulse was not used for these passes.

These passes afforded us the opportunity to observe the Ka-band performance over several passes under nearly identical conditions with the weather as the only variation. We could not do this with our dedicated Ka-band passes as consecutive passes over the same station were several weeks apart. Furthermore, we needed to verify many different functions with the dedicated Ka-band passes; thus the link configuration changed from pass to pass.

The shadow passes also allowed the gravity mapping team to obtain simultaneous X-band and Ka-band Doppler data to see whether or not Ka-band could be used to enhance gravity mapping of Mars. The SSNR data obtained during these passes were the first data from operational DSN stations that indicated that the spacecraft is producing adequate Ka-band EIRP. The data from DSS-34 were especially indicative of this (see Fig. 4 for example).

During some of these passes, slowdowns in the monitor data similar to those described in the previous section were also observed.

C. Shadow Passes Around Gravity Calibration 2

For these passes, the Ka-band team used the opportunity that the spacecraft was being continuously tracked from 26 December 2005 through 4 January 2006 to have the spacecraft transmit telemetry on the Ka-band link whenever the spacecraft was being tracked by a Ka-band capable antenna. The Ka-band configuration for these passes was identical to the X-band configuration, i.e., 550 Kbps concatenated coded data (1.1 Msps, 480 Kbps information data rate) with a 72-deg modulation index and ranging modulation index set to 17.5 deg. These passes were also important in that the distance from the spacecraft to Earth for these

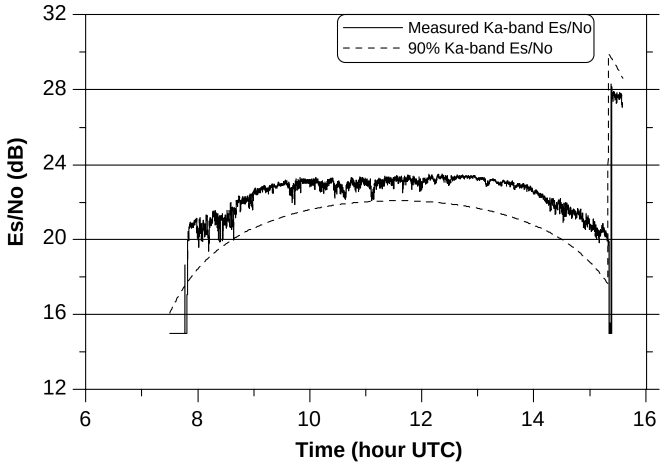


Fig. 4 Day 05-321 DSS-34 Ka-band E_s/N_0.

passes was approximately the same as the distance from Mars to Earth at their closest approach [approximately 0.6 astronomical units (AU)]. Therefore, the results from these passes are directly relevant for performance of end-to-end Ka-band link during the PSP.

Even though monopulse was not required for these passes, the stations used this opportunity to activate the monopulse to track Ka-band with varying degrees of success. For example, DSS-34 and DSS-55 successfully operated the monopulse on every one of their passes, whereas the only time DSS-25 used its monopulse, the system malfunctioned (see Fig. 5).

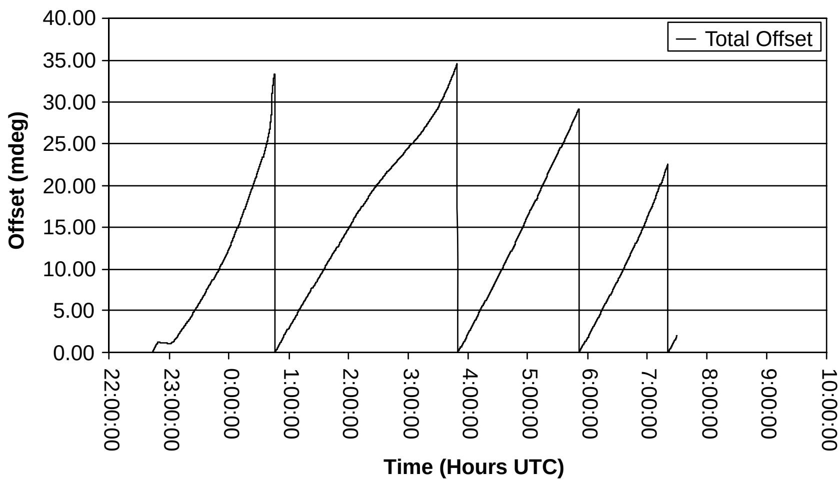


Fig. 5 Day 05-360, DSS-25 monopulse offsets.

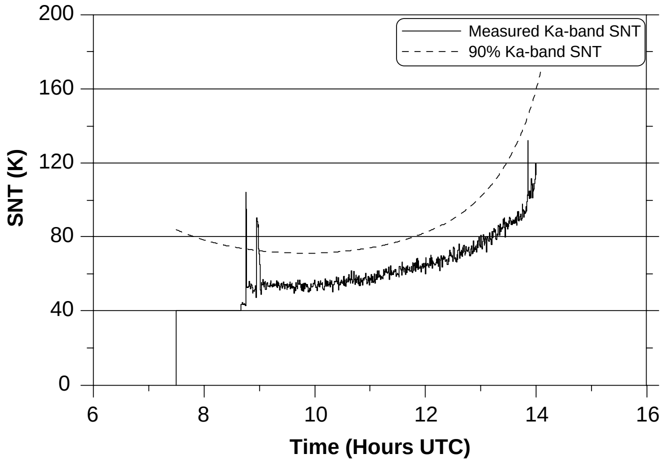


Fig. 6 Day 05-360, DSS-34 SNT (monopulse operational).

There were two important observations made during these passes. The first observation is that the SNT measurements for Ka-band seem to be accurate even at the shortest possible Mars-Earth distance, provided that the monopulse is working properly. This is indicated by the fact that when the monopulse is not working, the reported SNT values have larger fluctuations (see Figs. 6 and 7). The second observation is that under good weather conditions, the Ka-band link could outperform the X-band link (see Fig. 8). This means that Ka-band has the potential to return as much data as X-band for MRO and could be considered a viable back up for the X-band system with about a third of the X-band's transmitted power.

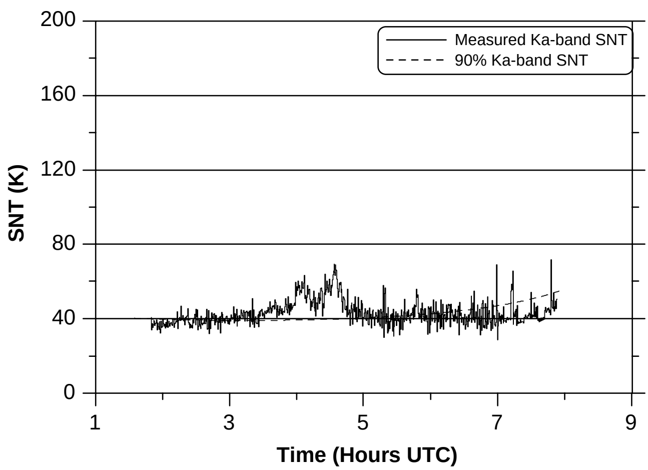


Fig. 7 Day 06-003, DSS-26 SNT (monopulse not used).

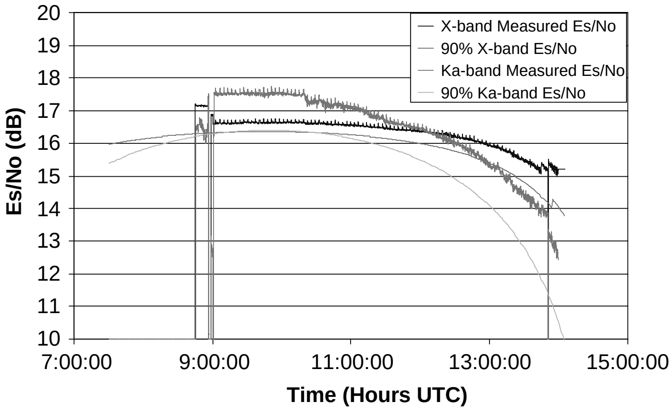


Fig. 8 Day 05-360 DSS-34 Ka-band and X-band E_s/N_0. (See also the color figure section starting on p. 645.)

Again during some of these passes, slowdowns in the monitor data updates were observed. This indicates that there is a recurring problem in the DSN with regard to delivery of the monitor data from the DSN complexes to JPL.

IV. Evaluating the Spacecraft EIRP

During the cruise phase, MRO presented challenges not normally associated with planetary missions because of its very high received downlink signal power. Because the high received downlink power caused errors in SNR and SNT estimates, it was not clear whether or not the high-gain antenna (HGA) calibration was successful. This also pointed to the fact that the DSN does not have a standard procedure for directly measuring the spacecraft received power. For power measurements, DSN relies on calculations based on SNR and SNT measurements. Because the SNR and the SNT estimates were affected by the high received downlink signal power, the DSN could not make an accurate estimate of the spacecraft received power and the spacecraft Ka-band EIRP at the beginning of the cruise.

Because the initial SNR measurements during the first Ka-band pass were lower than expected and initial measurements at DSS-13 indicated that the spacecraft EIRP was 5 dB below prelaunch measurements, a mission change request (MCR) was affected to leave the MRO Ka-band on in carrier-only mode for most of the cruise. This allowed us to develop methods for measuring the spacecraft EIRP directly without relying on a ground receiver. In the following we discuss the original HGA calibration and activities at DSS-13 [10] and at the 6-m array breadboard antennas at JPL Mesa.

A. High-Gain Antenna Calibration

The MRO project conducted an HGA calibration on 9 September 2005 (day 05-252) over Madrid, Spain. This calibration was intended to obtain the boresight of both the X-band and the Ka-band for the spacecraft HGA. For this purpose,

point-and-slew patterns at two different attitudes were used with a line separation of 0.1 deg. The Ka-band 3-dB beamwidth is only 0.18 deg; therefore, no beam pattern could be obtained from these scans. Instead a centroid approach was used to obtain the Ka-band boresightof the antenna. Unfortunately during this pass, the weather was bad; therefore, the SNR measurements fluctuated. In addition, not all of the spacecraft high-rate gimbal position data were transmitted after the calibration activity; therefore, the gimbal position commands rather than gimbal position knowledge were used for centroid calculations. The boresight calculations that were performed proved highly accurate. However, because of the coarseness of the calibration pattern and inaccurate SNR and SNT measurements, these calculations were deemed unreliable. Therefore, methods to validate them through direct measurements of the received downlink power were implemented at DSS-13 and the 6-m antennas at JPL Mesa.

B. DSS-13 Activities

As previously mentioned, because of the high spacecraft received power, the coarseness of the HGA calibration pattern, and errors in SNR estimates from the receivers, there were concerns as to whether or not the spacecraft was providing adequate Ka-band EIRP and whether or not the spacecraft antenna was on Earth-point. These concerns led to the development and exercise of measurement techniques that could be used for future high-power deep space missions. These techniques were exercised at DSS-13, a research and development (RD) 34-m beam waveguide antenna located in Goldstone, California, during several passes that spanned MRO's cruise phase.

Among the early activities performed at Ka-band (and X-band) at DSS-13 in September and October 2005 were several that involved checkout and calibration of various subsystems. These activities included characterization and measurement of SNT, measurement of system gain and linearity, measurement of the ground antenna efficiency using natural calibrator radio sources, exercising pointing techniques, measurement of MRO X-band signal strength, measurement of Cassini's X-band and Ka-band signal strength, Y-factor and follow-on noise temperature measurements, and attenuator adjustments. The initial calibration data and measurements of MRO Ka-band EIRP indicated that the DSS-13 Ka-band system was not fully optimal. This led to the need for further calibration and installation of additional equipment. A specialized filter was obtained to accept MRO's Ka-band 522 MHz intermediate frequency (IF) signal within a reasonably small bandpass (13 MHz noise-equivalent bandwidth) in the Ka-band chain. The center frequency and bandwidth of this filter were such that it could accept all of the power in MRO's Ka-band signal including carrier and telemetry. A series of additional activities and tests were performed in December at the station, including optimizing system response and linearity, adding amplification prior to the total power radiometer (TPR) and adjusting attenuation elsewhere in the system, and configuring the system to measure signal strength at radio frequency (RF) on day 05-350. IF measurements were performed with the TPR and the spectrum analyzer on day 05-350 as well as on other days.

Gain and linearity measurements were performed at the beginning of each session or when configuration changes were made. Calibrations were also performed periodically throughout each track in combination with the TPR or

spectrum analyzer measurements. A detailed discussion of the system calibration methodology is provided in [11]. Antenna efficiency measurements using natural calibrator radio sources [12] were performed at both X-band and Ka-band to evaluate the ground station gain correction in the estimation of the spacecraft EIRP.

A set of procedures was employed to measure signal strength using the TPR, and the station spectrum analyzers. The latter entailed measurement of carrier peak minus noise floor, or the carrier peak strength referenced to the LNA input using system gain calibrations.

The TPR method involves using the total power radiometer at DSS-13. TPR accepts two independent IF signal paths. In this case, for MRO one was from X-band right-circular polarization (RCP) IF and the other from Ka-band RCP IF. By peaking the antenna beam onto the signal using the boresight algorithm, the on-source system noise temperature is measured. Then, by moving off-source sufficiently, the background system noise temperature on the cold sky is measured. The difference between these two values is the SNT increase due to the spacecraft signal. On day 05-350, from about 7:00 to 10:00 coordinated universal time (UTC), the MRO spacecraft was tracked at X-band and Ka-band while measurements were being made at RF and IF. The system noise temperature increase due to the MRO Ka-band signal varied from 350 K to about 100 K during the pass (see Fig. 9). This is attributed to elevation dependence of both the atmospheric effects and the antenna efficiency at Ka-band. This system noise temperature increase was then converted to received signal power over the equivalent noise bandwidth (see Fig. 10 and also the color section of figures). Data that occurred during calibrations and boresight observations are removed from this figure. In Fig. 10, the measured received signal power using the TPR method is displayed in blue triangles. The red diamonds and green squares are spot checks using the spectrum analyzer method at IF and RF, respectively.

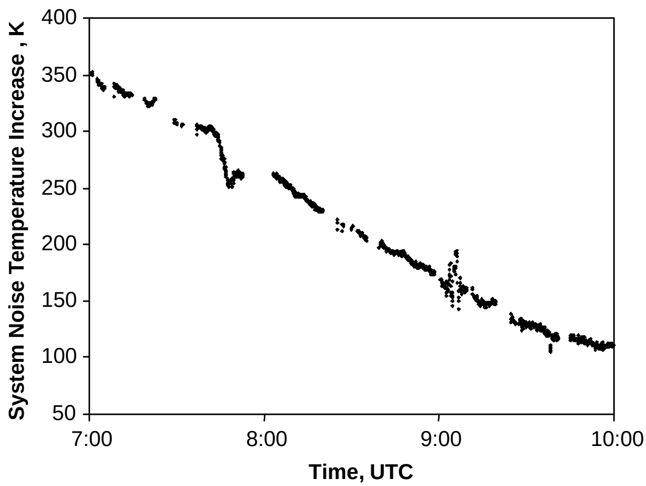


Fig. 9 System noise temperature increase due to MRO Ka-band signal at DSS-13 on day 05-350.

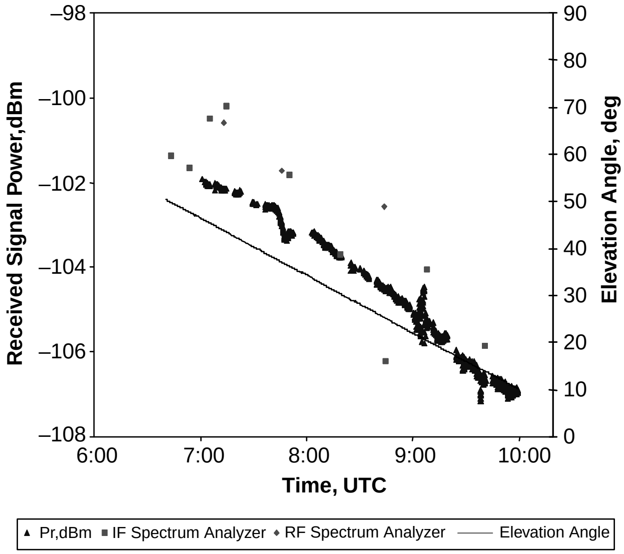


Fig. 10 Measured Ka-band signal strength received at DSS-13 on day 05-350. (See also the color figure section starting on p. 645.)

The received signal power in Fig. 10 was then converted to EIRP by using the link equation accounting for space loss, atmospheric attenuation, ground station gain, ground antenna mispointing, etc. The EIRP is referenced at the plane of the HGA and is displayed in Fig. 11 for day 05-350. The solid line denotes the predicted EIRP assuming the spacecraft HGA is perfectly on-point. The EIRP can

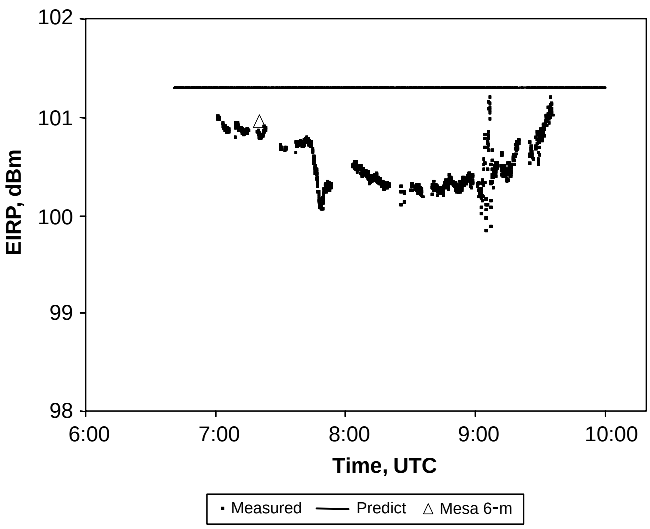


Fig. 11 MRO Ka-band EIRP on day 05-350 referenced at the spacecraft.

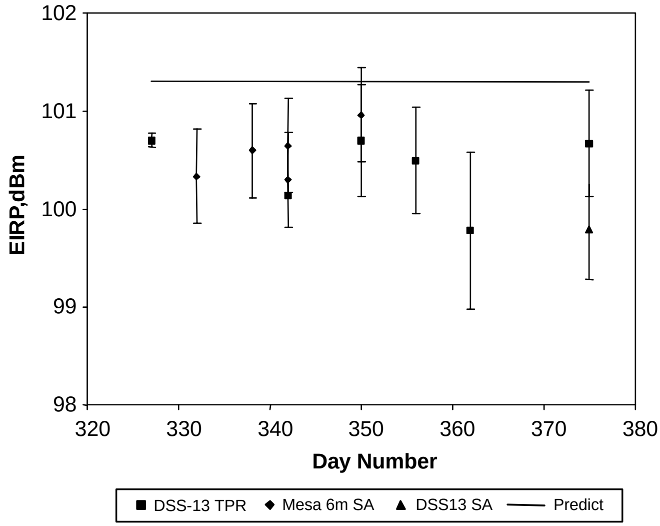


Fig. 12 MRO Ka-band measurement set at DSS-13 with JPL Mesa measurements and prediction.

be compared with prediction. This comparison provides an indication of how well the spacecraft is performing or of any problem such as spacecraft mispointing. The gain of the receiving antenna utilized curves derived from the antenna efficiency measurements of natural radio source calibrators as a function of station elevation angle. The atmospheric attenuation was calculated using a tip curve of off-source system temperature measurements and from surface meteorological parameters input into a weather model* when the weather was clear.

Upon inspection of Fig. 11, note that the measured EIRP is usually within 1 dB of the predicted and displays some interesting signatures that are not yet understood, but may be possibly attributed to some deficiency in the ground antenna efficiency model or possibly spacecraft motion. The uncertainty in Ka-band efficiency is less than 0.6 dB. Other error contributions such as those due to atmospheric attenuation are small.

In addition to the day 05-350 track, there were several other tracks that were conducted to measure the spacecraft's Ka-band EIRP (see Fig. 12). All of the DSS-13 measurements utilized the TPR or station spectrum analyzers (SA). Note that between days 05-325 and 05-360, there is good agreement among the measurements with all measurements indicating that the MRO spacecraft is on-point for Ka-band. The measurements conducted on 10 January 2006 (day number 375 on the graph, 06-010) was found to be about 4 dB low. This was due to a known spacecraft pointing offset related to a spacecraft safing event that occurred on 4 January 2006. As a result, a correction for spacecraft pointing using the known off-point angle was applied.

*The weather model is developed by Stephen Slobin of JPL.

The measurements depicted in Fig. 12 confirm that the MRO Ka-band EIRP measured at DSS-13 is consistent with the HGA being on-point, and lies within about 1 dB of predicted. The error bars on the DSS-13 TPR measurements represent the standard deviation of the measurements about the mean value over each pass, except for one pass for which the measurement was a spot check, in which case the error bar was assigned a value of 0.5 dB to conservatively account for unknown effects. For the 6-m antenna measurements, the error bars are 0.48 dB.*

The same procedures were exercised for the simultaneous X-band signal emitted by MRO, and the X-band EIRP was found to be consistent with prelaunch measurements.

C. 6-m Antennas

The two 6-m array breadboard antennas at JPL's Mesa are intended as test beds for the DSN large antenna array project. These antennas have a gain of 65.5 dB at Ka-band and have a cryogenically cooled feed producing a very low noise system. As of this writing, these antennas do not have any receiving equipment, but they do have a heterodyne mixer for downconverting the Ka-band RF signal to 1 GHz IF.

At these 6-m antennas, a spectrum analyzer attached to a laptop was used to measure the spacecraft EIRP. These measurements were based on an initial calculation of received carrier power P_c and using geometry and knowledge about the gain of the antenna to obtain the EIRP measurements. The carrier power is obtained in the following manner: first the antenna is off-pointed and the SNT is measured while the spectrum analyzer is connected to the IF. Matching the noise floor on the spectrum analyzer to the SNT calibrates the spectrum analyzer. Then the spacecraft signal is tracked and the relative P_c power is measured on the spectrum analyzer. From the calibration of the spectrum analyzer, the absolute value of P_c is calculated. Figure 13 shows a spectrum plot from day 05-333 (29 November 2005) from breadbaord antenna 1 and how the P_c was calculated from this spectrum.

When the spectrum analyzer is connected to a computer, P_c measurements could be made regularly from which the spacecraft EIRP could be calculated. Figures 14 and 15 show the difference between the measured EIRP and prelaunch EIRP for day 05-337 (3 December 2005) and day 05-347 (13 December 2005), respectively, obtained in this manner. As these figures indicate, the EIRP measurements match the prelaunch EIRP measurements very closely.

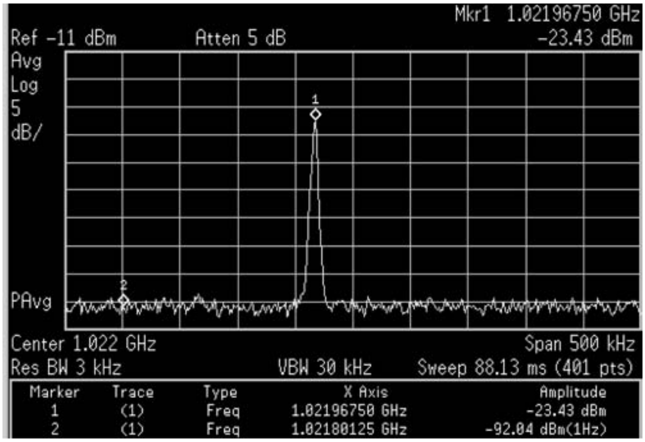
This method of measuring the spacecraft EIRP could be easily adopted by the DSN for making direct power measurement by connecting a spectrum analyzer to the IF patch panel at each complex.

V. ΔDOR Passes

The technique of ΔDOR has proved to be valuable for supporting spacecraft cruise navigation, especially for missions with tight targeting requirements at Mars. Spacecraft transmit tones, referred to as DOR tones, with a wide spacing from the carrier to enable these measurements. Today, measurements are made

---

*Communication with JPL's Michael Britcliffe.



- Tsys 36K
- Relative No (Marker 2)-92 dBm
- Relative Pc (Marker 1)-23.4 dBm
- Pc/No 68.6 dB-Hz
- Pc received (LNA)-114.4 dBm
- Pc predicted-114.16 dBm

Fig. 13 Spectrum plot and P_c calculation for 6-m antenna 2, day 05-333.

operationally in the DSN at X-band frequencies and provide an angular position accuracy of about 2.5 nrad. The deep space spectrum allocation and the restricted bandwidth of spacecraft transmitters at X-band limit the accuracy that can be achieved. The wider spectrum allocation for deep space tracking at Ka-band will enable an advance in ΔDOR measurement accuracy. Higher accuracy is needed to

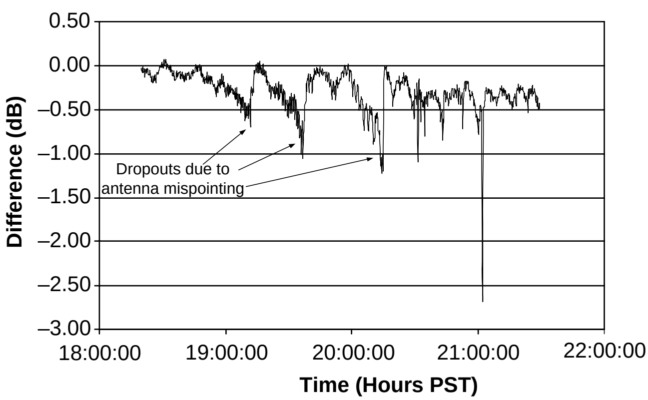


Fig. 14 Difference between measured Ka-band EIRP at the 6-m antenna and prelaunch Ka-band EIRP, day 05-337.

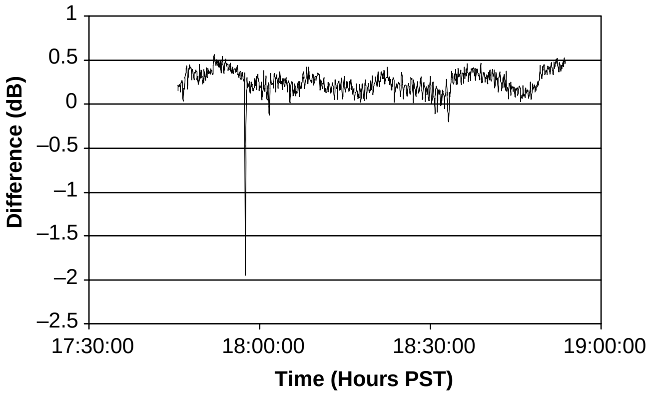


Fig. 15 Difference between measured Ka-band EIRP at the 6-m antenna and prelaunch Ka-band EIRP, day 05-347.

support future navigation challenges such as Mars landings or encounters with outer planet moons.

In a ΔDOR measurement, very long baseline interferometry (VLBI) systems are used at two stations to make high-rate recordings of signals from spacecraft and angularly nearby quasars. Antennas alternate between spacecraft and natural radio sources about 10 times in one hour. Radio source observations calibrate the system. For each source, the difference in signal arrival time between stations is measured and delivered to the navigation team. The MRO spacecraft emits DOR tones at both X-band and Ka-band. The DOR tone frequency is 19 MHz at X-band, yielding a spanned bandwidth of 38 MHz. The MRO transponder was designed for a DOR tone frequency of 76 MHz at Ka-band, providing a factor of four increase in spanned bandwidth. About 50 ΔDOR measurements were scheduled during cruise to support navigation. Of these, nine were selected to have dual-band X/Ka downlinks to demonstrate performance at Ka-band. Measurements were completed using the DSN 34-m BWG antennas that have X/Ka feeds. These measurements were the first ΔDOR measurements to be attempted at 32 GHz.

To prepare for these measurements, surveys were made of radio source flux, using the National Radio Astronomy Obsen (NRAO's) very large baseline array, at 24 and 43 GHz. Information from this survey was used to select radio sources to observe at 32 GHz. Whenever possible, sources are chosen to be within 10 deg of the spacecraft position, to have correlated flux of 0.4 Jy (jenskies) or higher, and to have structure index of 1 or 2. The structure index is a measure of compactness; a value of 1 or 2 implies that the typical delay error due to structure effects will be less than 0.04 ns. In the DSN, models for antenna pointing were improved to allow "blind" pointing to the coordinates of faint radio sources. The receiver used for VLBI data recording was modified to have a larger front-end bandwidth that allowed reception of the entire Ka-band spectrum allocation for deep space, including the received frequencies of the MRO DOR tones.

Data were successfully acquired at X-band and Ka-band for seven of the nine scheduled measurements. The measured data at Ka-band are in general agreement

with the measured data at X-band, and the precision of the Ka-band measurements is within expectations. The DSN receiving system worked well at 32 GHz. Antenna pointing was generally good, but some loss of signal power occurred because of errors in pointing that exceeded the very tight 4-mdeg requirement. One of the selected radio sources was found to have insufficient flux for use in these measurements, given the recording bandwidth that was used. Otherwise data acquisition was nominal.

Eventually, an operational system at Ka-band is expected to provide better accuracy than the current X-band system. The wider span of DOR tones improves group delay precision and reduces error due to dispersive instrumental phase. Both of these effects scale with spanned bandwidth. A higher data recording rate has been used at Ka-band, improving the precision of radio source delays. However, higher system temperature and lower quasar flux at Ka-band reduce the benefits of the previously mentioned effects. Ionospheric path delay is reduced by a factor of 15 at Ka-band relative to X-band. Radio source cores are more compact at higher frequencies, implying that, given sufficient source survey effort, an improved astrometric reference catalog could be defined using X/Ka data compared to that available today using S/X data.

The expected ΔDOR measurement accuracy is shown in Fig. 16 for three cases: MRO X-band data, MRO Ka-band data, and proposed future Ka-band data. Except for effects that depend on recording bandwidth, spanned bandwidth, and ionospheric path delay, all assumptions are the same for the three cases. For MRO, the

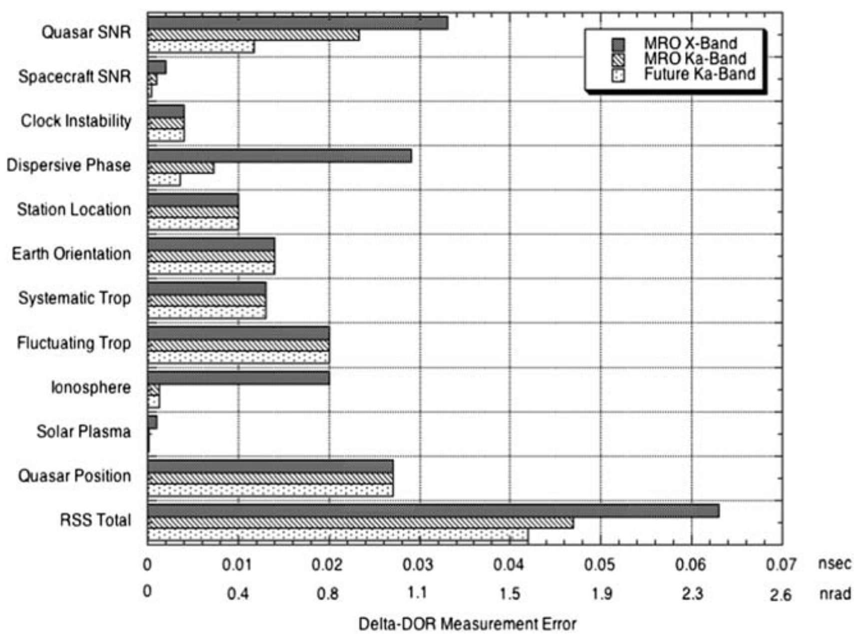


Fig. 16 Expected ΔDOR measurement accuracy for three cases.

quasar system noise error is reduced by $\sqrt{2}$ at Ka-band. This is the net effect of system temperature 2× higher at Ka, quasar flux 2× lower at Ka, spanned bandwidth 4× higher at Ka, and record rate 2× higher at Ka. For a future Ka-band system, the record rate could be further increased by 4× and the spanned bandwidth could be further increased by 2×. Dispersive instrumental phase is 4× smaller at Ka for MRO and would be 8× smaller for the future system. Ionospheric path delay is 15× smaller at Ka since charged particle delay is inversely proportional to the square of the frequency.

Figure 16 shows expected one-sigma accuracy for typical observing conditions for the three cases. Some improvement is seen at Ka-band for MRO, and further improvement in some error components is seen for the proposed higher bandwidth system at Ka-band. However, to take advantage of the reduction in the error components that is provided by transitioning to Ka-band frequencies, other work will be needed to realize a significant improvement in end-to-end system performance. An improvement in troposphere calibration will be needed. This could be achieved by making use of high-performance water vapor radiometers at each tracking station. Improvements in real-time knowledge of Earth orientation will be needed. This could be achieved by using a system similar to the ΔDOR system for making quick turnaround VLBI measurements of UT1. Finally, improvements in the global reference frame, including station coordinates and quasar coordinates, will be needed. This will require a measurement and analysis campaign, over several years, using radio source data at X-band and Ka-band. If development work is completed in these areas, then end-to-end system performance at Ka-band could improve to the 1-nrad level or better.

VI. Conclusion

As the results from the Ka-band activities during MRO cruise indicate, the spacecraft is fully capable of supporting the Ka-band demonstration during the PSP. Furthermore, while some minor issues with the monopulse antenna pointing and delivery of the monitor data still remain, these issues are expected to be resolved before the start of the PSP, and the DSN is expected to be completely ready for support of the Ka-band demonstration during the PSP. As a result of Ka-band cruise activities, MRO has set several milestones for planetary missions, including most amount of data returned in a single day (116 Gbits) and highest data rate (5.2 Mbps) ever. In addition, MRO has successfully demonstrated Ka-band ΔDOR operations during the cruise. Furthermore, as a result of Ka-band cruise activities, techniques for direct measurement of spacecraft received power have been developed.

Acknowledgments

This work was performed at Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. The authors would like to single out Ricardo Mendoza of JPL for his exemplary support of MRO Ka-band activities. Additional support for MRO Ka-band activities was provided by JPL's Dennis K. Lee, Michael Britcliffe, and Sander Weinreb. The authors would like to thank

Wallace Tai for programmatic support of this activity and James E. Graf, MRO project manager.

In addition, thanks go to JPL's Roy Gladden and the entire sequencing team as well as William Adams of Lockheed-Martin Corporation for supporting MRO Ka-band sequencing. The authors would also like to thank Paul Dendrenos for providing support in conducting tests and measuring RF system parameters and working many late shifts at DSS-13 on a moment's notice; Bill Lake for assisting in DSS-13 preparation activities, providing real-time support and working late shifts on a moment's notice; Ron Littlefair of DSS-13 for addressing MRO printing issues and performing requested tasks on short notice, and Bob Rees for coordinating DSS-13 station support.

We would also like to thank other JPL participants at DSS-13 during the 05-350 pass: Brad Arnold, Manuel Franco, Sam Zingales, and Stanley Butman. Also thanks go to JPL's Andre Jongeling for his superb work on the wideband VLBI science receiver (WVSR). The authors also appreciate the assistance and feedback provided by Dolan Highsmith of the MRO Navigation Team; Sami Asmar, Elias Barbinis, Daniel Kahan, and Kamal Oudrhiri of the Radio Science Systems Group; and Alex Konopliv of the MRO Gravity Science Team.

In addition, we appreciate the substantial help provided to us by Ronald Creech and Michael Wert of ITT technologies. Further thanks go to Philip Good of Infiniti Engineering, Inc., for support of HGA calibration activities; to JPL's Chau Buu, J. Andrew O'Dea, and Jeff Berner for support with DSN receivers and the monopulse; and to Miles Sue and the entire JPL Spectrum Management group for evaluating feasibility of BPSK use with X-band during the cruise and to Steve Keihm of JPL for providing advance water vapor radiometer support.

References

[1] Shambayati, S., "Optimization of a Deep-Space Ka-Band Link Using Atmospheric-Noise-Temperature Statistics," *TMO Progress Report 42-139, July-September 1999*, Jet Propulsion Lab., Pasadena, CA, 15 Nov. 1999, pp. 1–16.

[2] Shambayati, S., "Maximization of Data Return at X-Band and Ka-Band on the DSN's 34-Meter Beam-Waveguide Antennas," *IPN Progress Report 42-148, October-December 2001*, Jet Propulsion Lab., Pasadena, CA, 15 Feb. 2002, pp. 1–20.

[3] Shambayati, S., "On the Use of W-Band for Deep-Space Communications," *IPN Progress Report 42-154, April-June 2003*, Jet Propulsion Lab., Pasadena, CA, 15 Aug. 2003, pp. 1–17.

[4] Davarian, F., Shambayati, S., and Slobin, S., "Deep Space *Ka*-Band Link Management and Mars Reconnaissance Orbiter: Long-Term Weather Statistics Versus Forecasting," *Proceedings of the IEEE*, Vol. 92, No. 12, Dec. 2004, pp. 1879–1894.

[5] Shambayati, S., Davarian, F., and Morabito, D., "Link Design and Planning for Mars Reconnaissance Orbiter (MRO) Ka-Band (32 GHz) Telecom Demonstration," *IEEE Aerospace Conference*, Big Sky, MT, March 2005, pp. 1559–1569.

[6] Lock, R., and Sharrow, R., "Mars Reconnaissance Orbiter Mission Plan," Rev. C, Jet Propulsion Lab., JPL D-22239, Pasadena, CA, July 2005 (for JPL internal use only).

[7] Lee, D. K., "MRO Telecom Design Control Document," Rev. A, Jet Propulsion Lab., JPL D-22724, Pasadena CA, July 2005 (for JPL internal use only).

[8] Slobin, S. D., "DSMS Telecommunications Link Design Handbook, Module 104, Rev. B: 34-m BWG Stations Telecommunications Interfaces," Jet Propulsion Lab., JPL 810-005, Pasadena, CA, Aug. 2005, http://deepspace.jpl.nasa.gov/dsndocs/810-005/104/104B.pdf.

[9] Morabito, D. D., Mendoza, R., Highsmith, D., Border, J., and Shambayati, S., "Analysis of Ka-Band Radiometric Data Acquired During Cruise of the Mars Reconnaissance Orbiter," *Proceedings of the 12th Ka and Broadband Communications Conference*, Naples, Italy, Sept. 27–29, 2006.

[10] Morabito, D., Lee, D., Franco, M., and Shambayati, S., "Measurement of Mars Reconnaissance Orbiter Equivalent Isotropic Radiated Power during Early Cruise," *Interplanetary Network Progress Report 42-168*, Jet Propulsion Lab., Pasadena, CA, 15 Feb. 2007, pp. 1–19.

[11] Stelzried, C. T., and Klein, M. J., "Precision DSN Radiometer Systems: Impact on Microwave Calibrations," *Proceedings of the IEEE*, Vol. 82, No. 5, May 1994, pp. 776–787.

[12] Morabito, D. D., "The Characterization of a 34-Meter Beam-Waveguide Antenna at Ka-Band (32 GHz) and X-Band (8.4 GHz)," *IEEE Antennas and Propagation Magazine*, Vol. 41, No. 4, Aug. 1999, pp. 23–34.

This page intentionally left blank

Chapter 16

Perspective on Deep Space Network System Performance Analysis

Timothy Pham*

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

I. Introduction

RECENTLY, the Deep Space Network (DSN) re-instituted a formal activity on system performance analysis. This activity was previously practiced until the 1980s. It was de-emphasized in the 1990s because of a change in programmatic priorities. However, in late 2003 and early 2004, the DSN was tasked to support a series of mission-critical events. Among these were the landing of two Mars Rovers (Spirit and Opportunity), the orbit insertion of the Mars Express mission, the Wild-2 comet encounter by Stardust, the launches of the Deep Impact and Messenger spacecraft, and the Saturn orbit insertion of the Cassini mission. To adequately support these critical events, DSN equipment needed to be in the best possible operating condition. The need to monitor and analyze system performance had always been important, but it became even more so.

In 2004, a formal programmatic effort on performance analysis was re-established. The key objectives are to 1) ascertain the system operational readiness on a continual basis; 2) assess whether the DSN is meeting its service commitments to mission customers; and 3) monitor system performance margins. Included in this effort is the identification of performance weak links. The data are used to issue recommendations for improvement.

Figure 1 provides the context of performance analysis within the product-development life cycle. To provide long-term telecommunications services to missions, the DSN program must be concerned with the following three aspects: planning for future capabilities, implementing approved capabilities, and operating the network daily to return data to mission users. In the planning phase, the feasibility assessment determines if the requested capability is possible within the

*DSN Chief System Engineer.

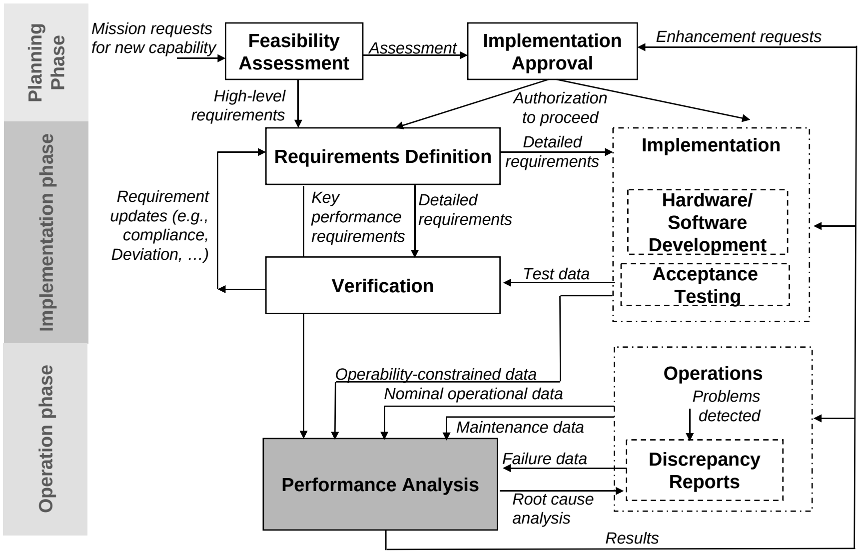


Fig. 1 Context of performance analysis within DSN development and operation cycle.

technical and resource constraints. The approval then starts the implementation. After the hardware and software are built, the equipment undergoes acceptance testing against the requirements. Any deviations are noted and put forth for consideration in future upgrades. Deviations from nominally expected system behavior, which may cause operational errors, are used in the analysis of failures. Upon successful acceptance testing, the new system is committed to operations. The performance analysis activity discussed in this chapter starts after equipment becomes operational. The aim is to ensure that all DSN equipment perform at the same level as when they were committed to operations. By use of analysis, operations staff can make any necessary corrections to maximize system performance. For changes that require upgraded hardware or software, the analysis feeds back into the planning phase, thus, closing the full cycle of planning, building, and operating.

This chapter presents the current progress the DSN has achieved in performance analysis. It first describes the functions and processes within this activity. It then addresses the metrics on data quantity, quality, continuity, and latency (QQCL) commitment to mission users. In addition, the analysis also focuses on other metrics that affect the link performance [such as antenna pointing; system noise temperature; Doppler accuracy; referenced frequency and timing synchronization, wide-area network loading; configuration setup time; and reliability, maintainability, availability (RMA) figures of merit]. The importance of these metrics, relevant to the mission operations, is explained. The currently observed performance, available margin, and trends for certain relevant metrics are presented. Relevant lessons learned through this effort are also highlighted.

II. Structure and Processing

Figure 2 shows the inputs, the outputs, and the functions within the performance analysis. There are three sets of inputs, operational data, behavioral data, and reference-model data. The operational data are performance data from spacecraft tracking. This data set constitutes the majority of information being processed. The reference model represents the expected performance under different operating conditions. They are the benchmarks to which the observed data are compared. The behavioral constraints supplement the reference model on operability aspects. They help to establish an understanding of problems related to operational procedures.

The operational data include discrepancy reports, link monitor logs, support data, and equipment maintenance records. Each is described in more details in the following:

1) Discrepancy reports identify any data outage that occurs within a tracking session. This data set identifies the components at which errors occurred (e.g., antenna servo controller or ranging processor). Other identifiers such as tracking antenna, supported mission, severity of data outage (e.g., lost, recoverable, or degraded) are also included.

2) Link monitor logs provide a rich set of information for mining performance metrics. The logs include key parameters related to the link performance (such as measured bit/symbol rates, signal-to-noise ratio (SNR), system noise temperatures, received frequencies, and antenna pointing correction). The logs also contain alarms and warnings issued by various subsystems. This information is important in performing failure diagnostics. The logs further capture the transactions when each link is set up, thus allowing derivation of metrics such as antenna setup time and signal acquisition time.

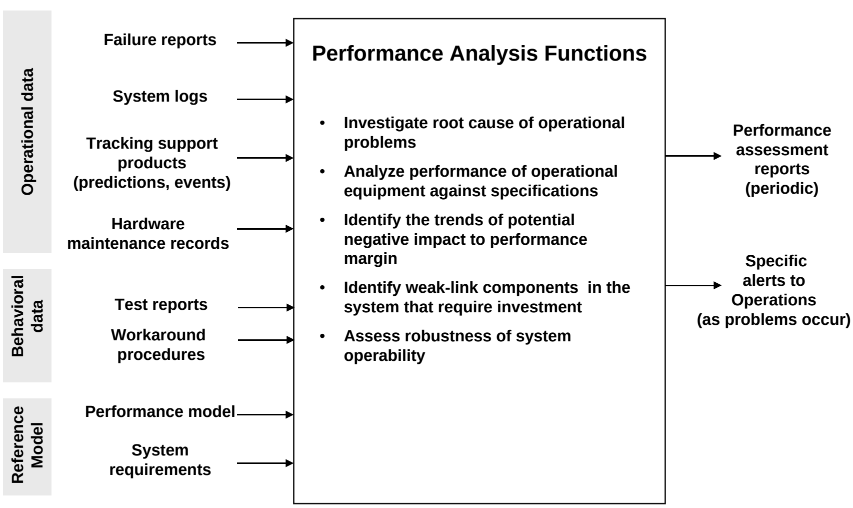


Fig. 2 Context of performance analysis functions.

3) Support data are the products needed for tracking spacecraft. They include the tracking schedule, pointing and frequency predictions, and expected telecommunications configurations (e.g., symbol/bit rate or SNR). The performance analysis uses schedule information to determine the expected tracking hours, from which service or system availability is computed. The prediction data are used to support failure diagnostics.

4) Hardware maintenance records identify necessary works done on the equipment to keep them functional. The records cover both preventive maintenance and unplanned repairs. From this data set, one can assess the maintenance effort in providing the services. The data identify what equipment requires the most care, thus directing recommendations for possible replacement. From these data, one can also determine the mean time to repair of failed equipment. The information provides insights on how well the equipment was designed in terms of maintainability focus and how proficiently the maintenance team performed the repairs.

The reference models represent the specifications to which actual performance is compared. There are performance models for various signal conditions (e.g., receiver loss as a function of SNR and tracking bandwidth, or system noise temperature as a function of elevation). Also included is a set of requirements for various capabilities.

The behavior data include test reports and anomalies identified at the time of new equipment delivery. There are also workaround procedures for items that deviate from the normal operation. These data establish the degree of system operability. A system with many anomalies and workarounds would likely have more operational errors. This is especially true when the operators are not the same ones who designed the system.

There are two main activities within performance analysis, as indicated in Fig. 3:

1) Root-cause analysis. This activity enables a true characterization of the problems, in particular for those that exhibit different symptoms. This process corrects for any incorrect attribution at the first recording due to limited understanding at the time when the problem surfaced. The root-cause analysis also directs attention to the common problems, thus heightening the chance for their correction.

2) Assessment. The system operating performance is evaluated against the requirements and reference models. The data quantify whether the DSN is meeting its commitments to mission users, how much margin is there, and what the future trends look like. It identifies the weak links that merit investment. A weak link could be one piece of equipment near or past its designed life cycle and requiring a great deal of maintenance attention. It could also be a software module with a persistent operability problem that could certainly benefit from an upgrade.

The analysis results are fed back to the DSN engineering and management so that appropriate corrective action can be taken. Engineering change requests for improvement could be submitted and funding solicited. However, when the detected problem poses significant impact and if it can be resolved within the current resource, the finding is relayed to station maintenance personnel for immediate resolution.

Figure 4 shows a more detailed view of the processing within performance analysis. First, data are captured and archived for possible reprocessing. Data

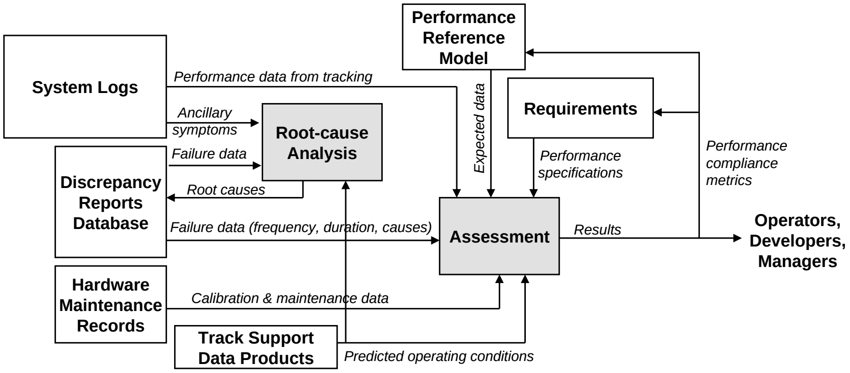


Fig. 3 Key processing functions.

extraction and mining then follow. The assessment can be grouped in four flavors: data QQCL, link performance and margins, RMA, and design robustness. Trending analysis allows for projection of future performance.

The performance analysis team has developed toolsets to automatically extract the data, plot them, and compute basic statistics (such as average and standard deviation). This enables the engineering team members to focus on the high-level analysis. The areas of QQCL, link performance, and RMA have been established, as discussed in Secs. III, IV, and V, respectively. Historical trending analysis is available, but much of projection of future performance is still under work. Assessment on system design robustness is still forthcoming. These last two items will not be further discussed in this chapter.

III. QQCL Assessment

Some of the goals set forth by the performance analysis team are as follows:

1) Institute a DSN-internal capability for determination of QQCL metrics. Until now, the DSN has only tracked data return via a time-based availability metric. This metric is computed as a ratio of non-outage time over scheduled time. There

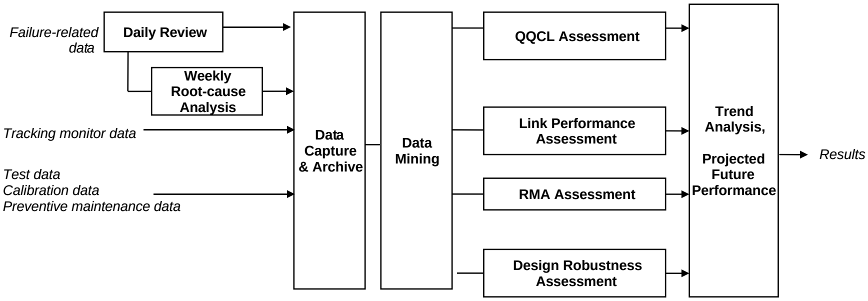


Fig. 4 Performance analysis internal processing.

has not been DSN accounting at frame level. Such data do exist, however, as accounted for by the mission data management teams.

2) Correlate the quantity/quality metrics against the expectation. The model for expected data return needs to go beyond just simple calculation of duration of tracking schedule, i.e., from beginning of track to end of track time. It needs to account for the expected signal loss within the window that is outside of ground system control (such as occultation when a spacecraft passes behind a planet and a change in spacecraft downlink frequency, which requires a signal reacquisition when it switches between one-way noncoherent mode to two-way coherent with an uplink).

A. Quantity Assessment

The DSN commitment for telemetry, tracking, and command data delivery is 95% for nominal operations and 98% for critical operations. Higher performance for critical operation is typically achieved with additional staffing on standby support and with redundant equipment. The data presented next are for most, and possibly all, nominal operations.

A sample of telemetry data return is shown in Fig. 5 [1]. It shows the frame return for Cassini and Mars Reconnaissance Orbiter (MRO) missions in roughly February 2006. The quality figure is computed as the percentage of good frames over the total frames captured and delivered.

Attempting to quantify metrics for command data turns out to be problematic. The challenge is that the expected command link transmission units (CLTU) cannot be sufficiently derived from the current expected sequence of events (which is contained in the support data product; see Sec. II). In a normal command session, the mission operation team has much flexibility in radiating a small set of CTLUs. For an 8-h track, the command session may take only half an hour; thus, it could occur at any time within the track. Therefore, the scheduled time from beginning to end of track does not represent the amount of transmitted data. In another scenario, the mission operation team could also choose to delay the transmission until the next tracking session if a problem is encountered, assuming the command session is not time sensitive. These kinds of decisions are often made in real time, and unfortunately the information is not readily available for the analysis effort.

To work around that difficulty, the performance analysis team relies on failure-indication metric, such as the event of “Command link transmission unit (CLTU) session abort.” The track without abort represents a successful track. A sample is shown in Fig. 6 [2]. Note that the presence of an abort event just indicates that a problem was encountered in the uplink at some point during the pass. It does not

	Day of Year	Total Frames Delivered	Out-of-Sync Frames	Quality, %
Cassini	026-064	1165475	0	100
MRO		1994844	145	99.99

Fig. 5 Sample of telemetry frame accounting.

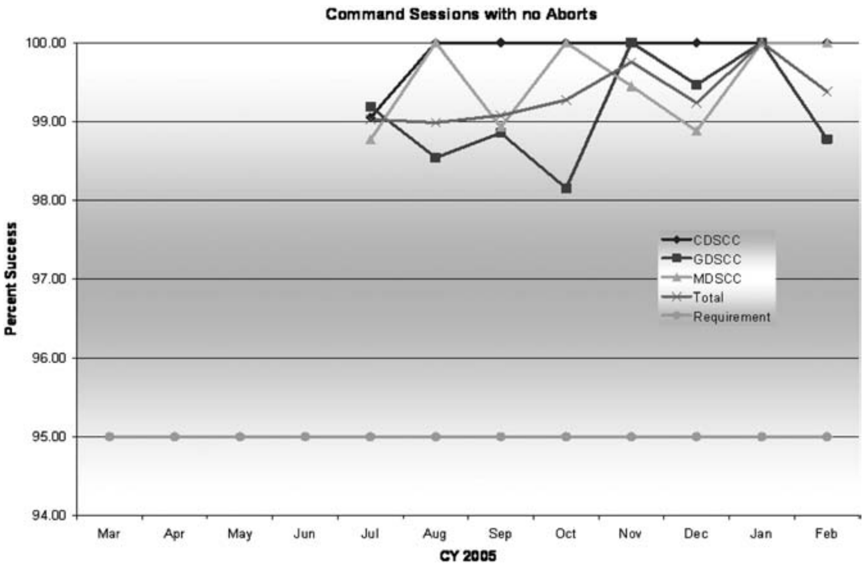


Fig. 6 Figure of merit for command quantity/quality metrics. (See also the color figure section starting on p. 645.)

necessarily imply a complete uplink failure for that pass because a new CLTU session could have been successfully re-established soon after. Thus, the presented statistics provide a lower bound of system availability.

In Fig. 6, statistics of successful uplink (without error encountered) are shown for all three Deep Space Communications Complexes (DSCC) at Goldstone, California (GDSCC), Canberra, Australia (CDSCC), and Madrid, Spain (MDSCC). The network average is presented by the "Total" data. The achieved performance from March 2005 to February 2006 is quite high, above 99%, relative to the 95% requirement.

B. Quality Assessment

This assessment focuses on the quality of telemetry frames produced under nominal conditions (e.g., sufficient link margin). The DSN commitment is a maximum frame error rate of 10^{-6} for the concatenated convolution and Reed Solomon code. For turbocode, the maximum error rate is 10^{-4} for long frames (8912 bits) and 10^{-5} for short frames (1784 bits). Figure 5 shows the DSN support to Cassini mission meets the commitment for convolutional code. For MRO, the 99.99% return translates to a frame error rate of 10^{-4}, which meets the commitment for long turbo code.

C. Continuity Assessment

The DSN commitment for telemetry data continuity is maximum 8 gaps per 10,000 frames. A gap is defined as a set of consecutive undecoded frames. At the

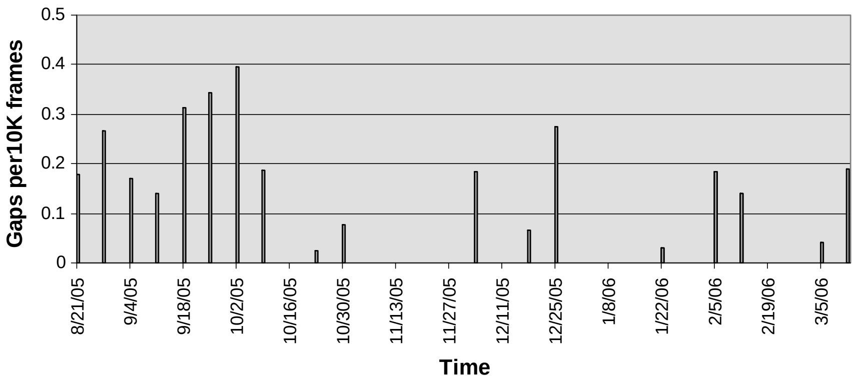


Fig. 7 Observed gap statistics from MRO mission.

present time, the DSN-internal metrics on gaps are still under development. A sample of actual statistics for the MRO mission produced by the external Mission Data Management Team is shown in Fig. 7 [3]. The observed gap is significantly less than the requirement.

D. Latency Assessment

Different telemetry data have different latency needs. For spacecraft engineering data, the mission would want to have the data delivered immediately for timely monitoring and control purposes. For mission science data, the delivery can be longer because the urgency is less critical. The DSN normally commits to a delivery time of 10 s for spacecraft engineering data and within 24 h for mission science data. The throughput constraint is caused by limited bandwidth in the wide area network that connects the three tracking complexes and the Jet Propulsion Laboratory (JPL), where missions obtain their data. The latency is defined as the difference between the Earth-received time recorded at the tracking complex and the arrival time at the Network Operation Center at JPL. Figure 8 [4] shows the averaged and maximum latency of spacecraft engineering telemetry for the Solar and Heliospheric Observatory (SOHO) mission in February and March 2006. The typical delay is less than 1 s. Similarly, Fig. 9 shows the averaged and maximum latency of MRO's telemetry science data. The delay is typically a few seconds. Both data sets reflect good performance, well within the commitment.

IV. Link Performance Assessment

On link-related performance, some key parameters of interest are 1) antenna pointing accuracy, 2) system noise temperature, 3) doppler accuracy, 4) frequency and amplitude stability, 5) wide-area network (WAN) bandwidth loading, 6) synchronization of frequency and timing references, and 7) link configuration setup time.In the following sections, some preliminary findings will be presented. Because the analysis effort is ongoing, the presented data should be viewed as a

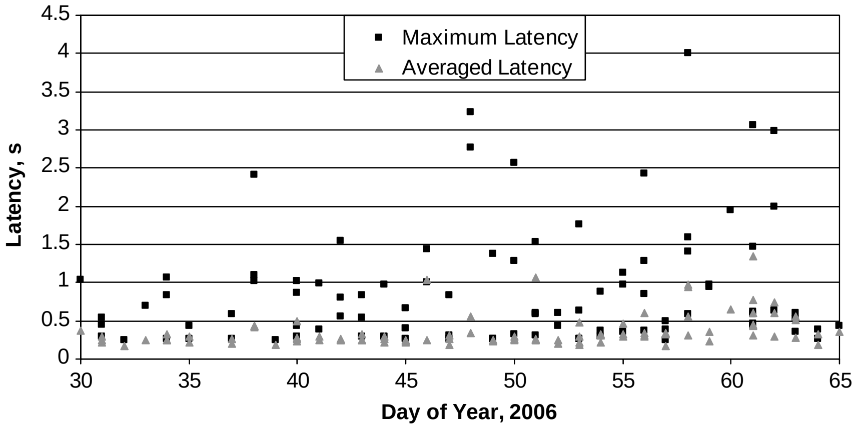


Fig. 8 Latency of SOHO mission's engineering telemetry.

snapshot of existing progress. The findings are not final, and many of the unexpected deviations will require further investigation.

A. Antenna Pointing

As the DSN operation is moving toward higher frequency Ka-band (32 GHz), the antenna beamwidth is becoming smaller and the antenna gain more accentuated. Thus, the precision in antenna pointing becomes more critical to realize the maximum gain. The requirement is 4 mdeg with program-tracked pointing without feedback (also known as blind pointing in DSN terminology), and 2 mdeg with active closed-loop monopulse pointing correction. Preferably, one would want to have closed-loop monopulse pointing correction all the time; however,

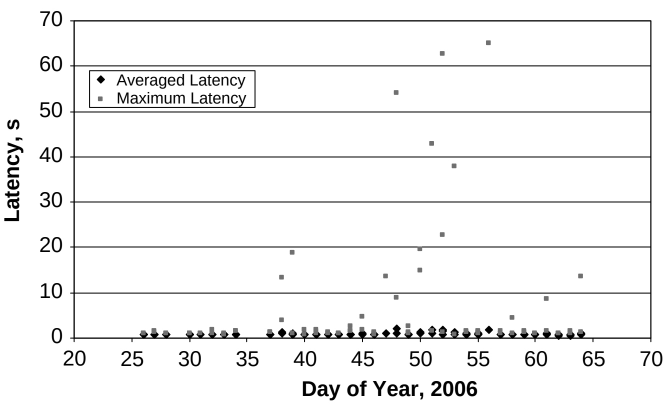


Fig. 9 Latency of MRO telemetry science data.

there are situations where this is not possible. Such is the case with the signal source being a planetary object. Its signal tends to be low and broadband. In contrast, a spacecraft signal has the advantage of being stronger and coherent within a narrow band. The program-tracked pointing is typically needed for delta differential one-way ranging (ΔDOR) application. Also, to some extent, the program-tracked pointing needs to be good enough to bring the antenna close to the targeted spacecraft so that monopulse tracking acquisition can be successful. Once acquired, the active monopulse tracking algorithm can keep the antenna on source.

Figure 10 [5] provides a sample of pointing errors at one of the 34-m antenna beam waveguide antennas. The errors are plotted in azimuth and elevation coordinates. While for the most part the pointing is less than 4 mdeg, there are areas where pointing error is significantly higher. Information such as this can be used to improve the pointing model, resulting in a better performance. Note that this data set only reflects a snapshot of performance at the current time. The antenna is in transition to using a better pointing model, which should lead to better performance.

B. System Noise Temperature

A sample of actual noise temperature is shown in Fig. 11 [6]. The system noise temperature typically varies as a function of tracking elevation. The variation is due to geometry and can be mathematically modeled with a relatively good accuracy. To simplify the analysis, one can reliably compare the actual and expected noise temperature near the zenith. In this region, the noise temperature dependency on elevation becomes minimal. Proper grouping of data is also necessary because the noise temperature is also dependent on the microwave configuration, e.g., operating frequency (S, X, Ka), diplexed vs listen-only, type of low noise.

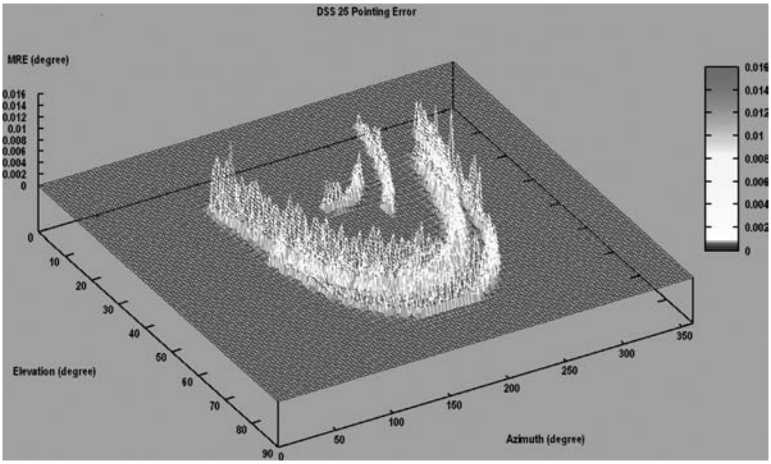


Fig. 10 Sample pointing accuracy of a 34-m beam waveguide antenna. (See also the color figure section starting on p. 645.)

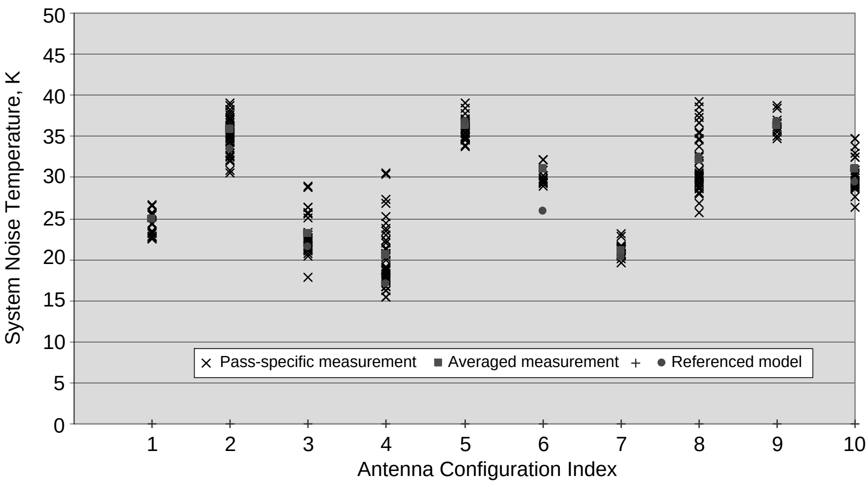


Fig. 11 Sample of observed system noise temperatures at Goldstone (near zenith). (See also the color figure section starting on p. 645.)

Presented in Fig. 11 under each configuration are per-pass measurements of system noise temperature (SNT), the average measurement for a configuration, and the expected noise temperature per model.

There are two unexpected observations. First, there are instances where the variation among SNT measurements within one configuration is much larger than expected. Second, there are cases where the average SNT is significantly higher than what the 90% weather model predicts. These inconsistencies are being investigated and hopefully will be resolved with future analysis.

C. Doppler Noise

Doppler noise affects navigation accuracy. A significant deviation from a typical performance level, and from the specification, would indicate a potential problem. Perhaps there is an error in the frequency references, or the signal being saturated, or the received signal is unstable. Figure 12 [7] shows a sample of the observed Doppler noise for Cassini mission. The Doppler noise drops at higher SNR, as one would expect. One also notices that there is an unexpected high noise level at 32 dB and at 40–50 dB SNR. This simple plot allows the team to focus on the unexpected conditions and to try to resolve the anomaly.

The observed Doppler actually contains three noise sources. Besides the ground system, it also includes the noise from spacecraft and from the traversed space media. Separating the ground system effect and comparing it against the ground system requirement will be quite a challenge.

D. Frequency and Amplitude Stability

Besides the telemetry, tracking, and command functions, the DSN also serves as a scientific instrument in radio science experiments. Performing a gravitation

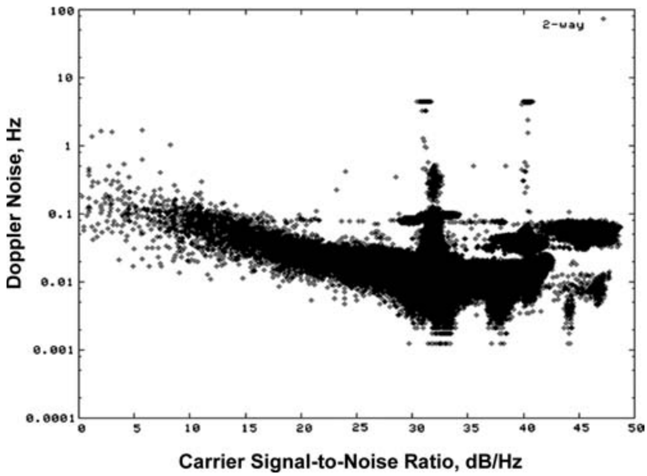


Fig. 12 Sample of Doppler noise observed with Cassini mission tracking.

wave search requires a system with very stable long-term frequency stability, on the order of hours that correlates to the round-trip light time of the experiment. In studies of a planetary ring or an atmospheric occultation, the physical characteristics of the ring or the atmosphere can be deduced from the amount of attenuation the signal undergoes as it travels behind the object of study. To support such an investigation, the DSN system needs to have good amplitude stability.

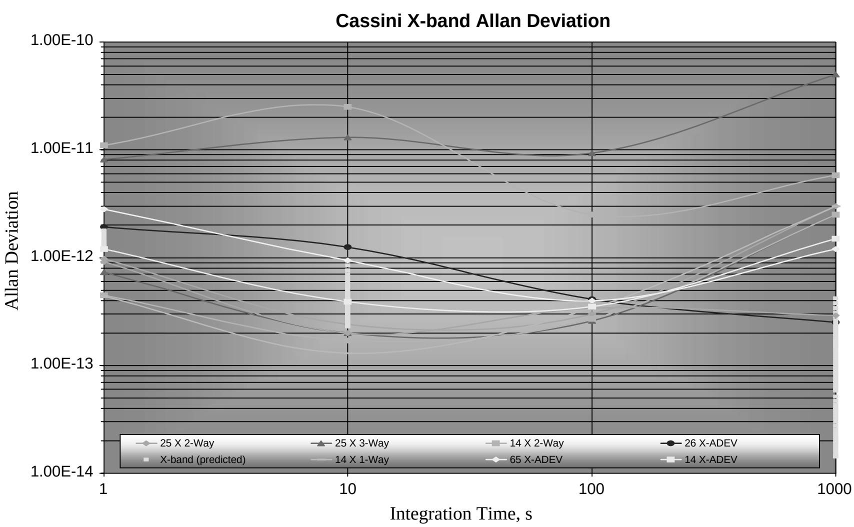


Fig. 13 Frequency stability between observed and model. (See also the color figure section starting on p. 645.)

Figure 13 [8] shows the actual vs modeled frequency stability. This model includes the effects of the ground system, the space media, and the spacecraft equipment. The space media effects include those from solar scintillation, the Earth troposphere, and the Earth ionosphere. The prediction range accounts for different observation geometries, which causes a variation in the media contribution at X-band. The actual data are measurements from different antennas at various times with the Cassini spacecraft. The observed Allan deviations exhibit a large variation for different tracking passes. For the most part, the observed stability at 1-, 10-, and 100-s integration approximate the model; however, the 1000-s integration data deviate significantly from the model. Two passes are atypical from the rest of the data set. More effort is required to understand the variability in the data and to resolve the discrepancy between model and observation.

Fig. 14 [8] shows the amplitude stability observed with the Cassini spacecraft at one of the 34-m beam waveguide antennas. The observed stability at X-band, based on the drift average, over 30-min duration, is 0.09 dB, relative to the 0.2 dB requirement for the ground system. Note that the observation data include the drift in both ground and flight equipment; thus it encompass more the ground specification.

E. Wide-Area Network Loading

In today's configuration, the DSN facilities are interconnected by a set of dedicated wide-area network connections. The connection is made of a few T1/E1 circuits. The bandwidth is limited, 4.5 Mbps for the Canberra and Madrid tracking complexes and 6.0 Mbps for the Goldstone tracking complex. It is important to monitor the network loading to gain an understanding on how heavy the traffic is and how much margin remains. This information also helps to validate the network traffic model. A good model, in turn, aids proper future planning.

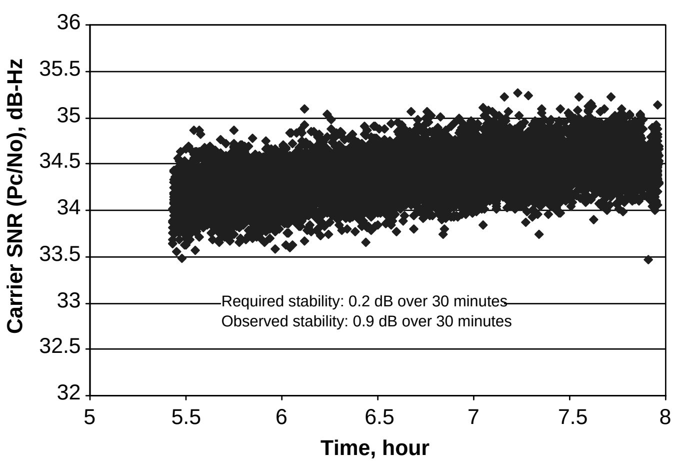


Fig. 14 X-band amplitude stability at a Goldstone 34-m antenna.

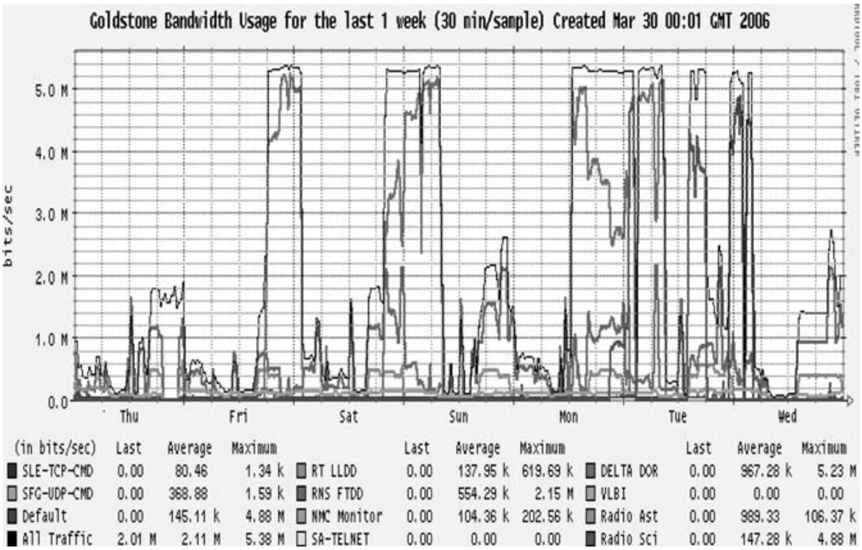


Fig. 15 Sample of network utilization. (See also the color figure section starting on p. 645.)

A sample of the network monitoring is provided in Fig. 15 [9]. Various types of data traffic (e.g., real-time and science telemetry, monitor, and ΔDOR) are shown over the course of one week. The total bandwidth loading is reflected by the "All Traffic" profile. On average, the loading is 2.1 Mbps, which is about 40% of the maximum capacity (5.4 Mbps at Goldstone). There are times when the maximum bandwidth is used, which usually coincides with the transfer of ΔDOR data. In general, that is not an issue because the ΔDOR data transfers always take as much bandwidth as the line offers; however, it is of lower priority compared to telemetry dataflow. Thus, in the presence of telemetry data, the ΔDOR transfer will be metered back to enable telemetry data delivery to meet its latency requirement. However, one would want to make sure that the line is not constantly booked at maximum level, as in the crowding condition seen in the later part of the week. If such a condition were to persist over a long period of time, an increase in the bandwidth capacity would be desirable. Fortunately, the DSN wide-area network is being upgraded, and soon the Goldstone link will be increased to 12 Mbps.

F. Time and Frequency Reference Synchronization

To support precise Doppler measurement for navigation purpose, the frequency and timing references need to be as close to the standard reference of the National Institute of Standards and Technology (NIST) as much as possible. The requirement for timing offset is 5 μs or less. Figure 16 [10] shows a sample of the timing offset at the Madrid tracking complex.

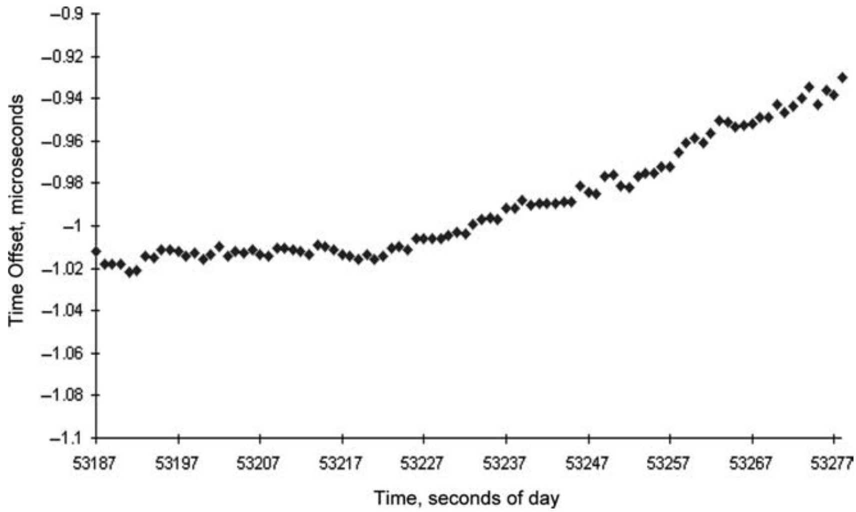


Fig. 16 Timing synchronization between Madrid DSCC and NIST.

G. Link Setup Time

Current DSN scheduling allows for a fixed time to set up certain services. This setup involves configuring equipment into the link, going through some internal validation, moving the antenna to point, and warming up the transmitter if the uplink is required. For telemetry service, 30 min is the booked setup time. If concurrent command and tracking services are to take place, the setup time would be increased by 15 min.

The monitoring of actual setup time helps to assess how well the DSN prepares for the track. From this analysis, one can identify possible bottlenecks in the process. The assessment offers an opportunity to increase the efficiency of network operation with a reduced setup time. The recommended time reduction, however, needs to be balanced between increased efficiency and possible increases in the failure of not being ready for the track.

Figure 17 [11] provides a sample of the setup performance for the months of January and February 2006. The data are normalized against the current allocated time. Thus, 100% indicates that the actual setup requires all of the allocation time. The data show that, on the average, operations personnel at Goldstone got the link equipment ready within 71% of the allocated time. About 90% of the time, operations personnel finish the job within the time allocation. The next analysis step is to assess the optimal reduction time, within an acceptable increase in failures.

V. Reliability, Maintainability, Availability Assessment

The assessment on RMA focuses on common causes of failures, as well as the frequency and impact of failures. In general, failures in the antenna and transmitter subsystems tend to result in long outages because they are dominated by large

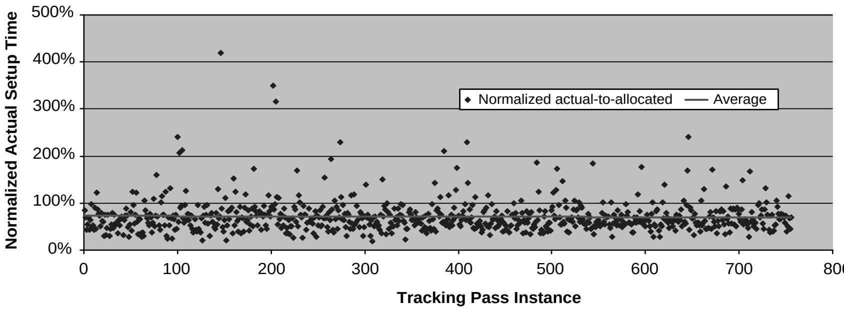


Fig. 17 Pre-track setup time, January through February 2006.

mechanical and high-power components, which require longer times to restore to service. Failures in other electronic-based, software-centric subsystems tend to be short, although they occur more often. Among the various antennas within the DSN, the 26-m antenna group suffers greater reliability problems because of older equipment. There also has been less investment made in this antenna subnet, as compared to the 34-m and 70-m antenna subnets. Another area that needs attention is the power generation and distribution equipment. Some power components are fragile, having well exceeded their design life span. Because power is required for operation of other subsystems, it is expected that proper investment will need to be made to shore up this infrastructure.

VI. Conclusion

This chapter presents the context of ongoing performance analysis activity that is taking place in the Deep Space Network. This activity includes root-cause analysis of failures to enable effective solutions. It also includes the performance assessment, aiming to establish how well the equipment is meeting its specifications and how much margin is available. The assessment establishes a baseline performance on data-return metrics and specific performance-related parameters that affect the quality of services provided. These include antenna pointing, system noise temperature, Doppler noise, frequency and amplitude stability, wide-area-network bandwidth loading, offset in frequency and time references, and the track setup time.

In general, the system exhibits good performance and meets the specifications. The averaged statistics on data QQCL, bandwidth loading, frequency and timing offsets, and tracking setup time show a reasonable margin. An area of concern is antenna pointing, particularly at Ka-band. More effort will be required to achieve a consistent performance as specified.

The performance analysis activity is still in the early stage of development. Additional effort is needed to improve the modeling to properly account for the observations. There remain many unexplored metrics. Thus, this work continues to be active and exciting in the near future. Looking back, some of the lessons learned are the following:

1) Accurate accounting of data return relative to the expectation is a challenge. More work is needed to establish the expected data return at the level of telemetry frames or transmitted command units. The information available in today's system, which is time based, does not present itself readily for this analysis.

2) Correct interpretation of the observed performance is crucial. Often, the specifications are for the ground system. The observation data collected from tracking spacecraft, however, include other non-ground effects, such as space media and the spacecraft. One has to be careful with the comparison. To the extent possible, one needs to create a model that accounts for all effects.

3) Special care is required in data processing. Rather than blindly processing the measurement generated by the system, one needs to verify that the generated data are valid. For example, in the case of system noise temperature analysis, the measurement accuracy is dependent on the SNR condition. A too-strong received signal may corrupt noise temperature measurements; thus, data would need to be excluded.

Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

The performance assessment presented in this report is a result of joint effort of several engineering and operation teams within the Deep Space Network (including the affiliated contractors). A special recognition is extended to Jason Liao in data extraction and mining that enabled assessment on key metrics of data QQCL, antenna pointing, and system noise temperature. The author would also like to thank the following team members at JPL for their leadership and contribution to this noble effort: Tim Cornish on uplink assessment; Andrew Odea on telemetry processing; Dong Shin and David Berry on tracking; Mick Connally and Alina Bedrossian on science data processing; Arthur Freiley, John Cucchissi, Paul Cramer, and James Bowen on antenna and microwave health status; Steve Slobin for antenna pointing and system noise temperature analysis; Mark Harris of Raytheon Australia for antenna pointing calibration; Ron Norman on service management and link configuration; Robert Tjoelker on the referenced frequency and timing; Jason Liao on the ground communications; John Luvalle of ITT for the general statistics on discrepancy reports; Luis Huertas of ITT, Hossein Hosseini, and Leslie White on facilities infrastructure. Last but not least, an unwavering programmatic support from Wallace Tai prompted the start of this activity and keeps it going.

References

[1] Shin, D., "Telemetry Performance Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 20 March 2006.

[2] Cornish, T., "Uplink Performance Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 20 March 2006.

[3] "Mars Reconnaissance Orbiter (MRO) Multimission Ground Systems and Services (MGSS) Packet Spot Check Summary," Multi-Mission Data Management Team, Jet

Propulsion Lab., Pasadena, CA, http://ddo.jpl.nasa.gov/ddo/dmt/ [retrieved March 2006].

[4] Liao, J., and Shin, D., "Telemetry Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 20 March 2006.

[5] Liao, J., "Antenna Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 19 May 2006.

[6] Liao, J., and Bowen, J., "Microwave Performance Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 20 March 2006.

[7] Berry, D., "Tracking Performance Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 7 Dec. 2004.

[8] Bedrossian, A., "Science Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 14 July 2006.

[9] Liao, J., "Ground Communications Assessment," JPL Internal Portal, Jet Propulsion Lab., Pasadena, CA, 30 March 2006.

[10] Tjoelker, R., "Frequency and Timing Reference Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 20 March 2006.

[11] Norman, R., "Service Management Assessment," JPL Internal Briefing, Jet Propulsion Lab., Pasadena, CA, 20 March 2006.

Chapter 17

Overview of JAXA Space Debris Surveillance Operations

Chikako Hirose* and Yoshitaka Taromaru†

Japan Aerospace Exploration Agency, Tsukuba, Ibaraki, Japan

Masaya Kameyama‡

Fujitsu Laboratories Ltd., Chiba-city, Chiba, Japan

Sadao Aoki§

Japan Space Forum, Chiyoda-ku, Tokyo, Japan

and

Syuzo Isobe¶

Japan Spaceguard Association, Shibuya-ku, Tokyo, Japan

I. Introduction

SINCE Sputnik 1 was launched on 4 October 1957, approximately 6000 satellites have been launched. With increasing launch and development of space activities, the space debris environment has also started to take shape. So far, nearly 30,000 objects including satellites, rocket bodies, and those fragmentations have been put into orbit. Twenty thousand have already decayed, whereas more than 7000 space debris objects are still drifting around the Earth. Depending on the solar activity, 150–500 space debris objects reenter the atmosphere each year. On the other hand, the number of reentering space debris objects are still limited, and referring to the database of U.S. Space Track [1], the total number of space debris objects has increased by roughly 800 in the past 10 years, even though this increase is becoming more gradual than it was in the 1970s or 1980s.

---

*Engineer, Office of Space Flight and Operations.

†Associate Senior Engineer, Office of Space Flight and Operations.

‡System Engineer, Science Solution Group.

§Assistant Manager, Space Utilization Promotion Department.

¶Director.

Under these circumstances, we can see clearly that we need to understand the distribution of space debris with further accuracy. The United States, Russia, and some countries in Europe have implemented the observation of space debris. However, very few countries have the facilities for observation. Japan Aerospace Exploration Agency (JAXA) started the space debris surveillance operation with radar and optical telescopes in recent years. The radar and the telescopes were built by Japan Space Forum (JSF) in cooperation with JAXA; the radar was built at Kamisaibara Spaceguard Center (KSGC) in 2004, and the telescopes were built at Bisei Spaceguard Center (BSGC) in 2000. These two systems are located in Okayama, the western part of Japan. The KSGC radar is the first radar system for space debris exclusive use in Japan. Tsukuba Space Center (TKSC) space-debris data-processing capability communicates with KSGC for requirements transmission and receiving the observation data. The BSGC has three telescopes: 25 cm, 50 cm, and 1 m. Our group uses 50-cm and 1-m telescope data by sending the observation requirements from TKSC.

We describe the capability and function of these systems in Sec. II and introduce the radar and optical observation concept and method in Sec. III. The observation result is discussed in Sec. IV. Based on these results, we overview the plan for future observation and at the end make conclusions.

II. Capability and Function

A. Radar System at KSGC and Data Processing Capability at TKSC

This radar adapts the flat active phased array (APA) as a pilot system [2, 3]. Figure 1 shows the KSGC radar, whose size is 3×3 m. It has 1395 transceiver modules (TRM) and transmits 70 kW as a peak level. Considering the skyline of the KSGC site, the elevation is fixed at 45 deg, and it can observe from 15 to 75 deg

Fig. 1 KSGC radar.

by electrical scanning. Regarding the azimuth scanning area, when we set the true north of KSGC as 0 deg, this radar can rotate mechanically 270 deg to both the east and west side and electrically it can scan ±45 deg in addition to the mechanical movements. As for the measurement accuracy, the angle is measured in less than 0.18 deg in azimuth by monopulse and 0.28 deg in elevation by sequential lobing.

The radar system has the capability to detect a 1-m-across sphere at the slant-range of 577 km, and the detection limit is 1350 km. This means that this system is aiming at the low-Earth orbital debris. The KSGC radar has a capability to track up to 10 space debris objects simultaneously once the radar detects the target with certainty. On the other hand, there is difficulty, caused by geographical disadvantage, in observing objects whose inclinations are less than 30 deg, since those objects pass rather far from KSGC.

At TKSC, we process the observation data for orbit determination and prediction. This system has two main characters. One is the capability of the automatic orbit determination and another is the automatic planning system for the best suitable observation to detect as many objects as possible simultaneously. The orbit determination is processed automatically by choosing data of three passes observed in the continuous five days. The observational requirements are sent from TKSC for the individual target based on the KSGC radar capability and conditions such as the azimuth rotation and beam direction control.

The plan position indicator (PPI) provides integrated operation, and operators can monitor the tracking status in real time at TKSC. Figure 2 shows the screen image of the PPI. Each line represents each trajectory of one space debris object and is drawn by each different color (see the color section of figures). In addition to the PPI, the real-time trajectory estimation program for space

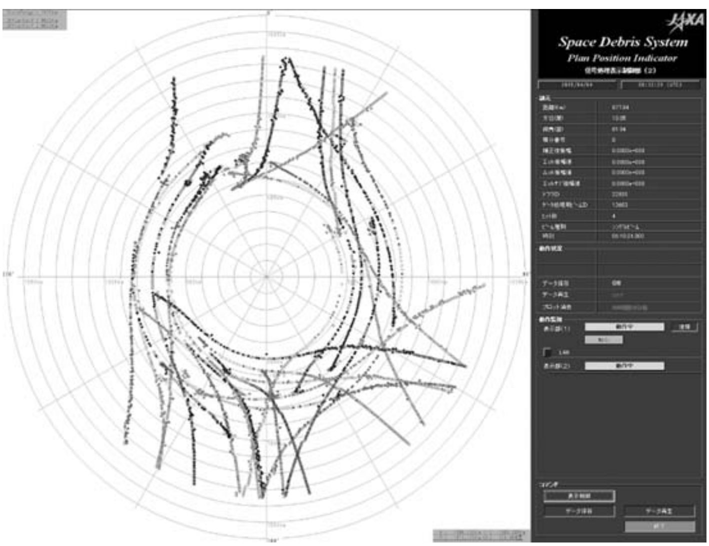


Fig. 2 Plan position indicator at TKSC. Each line represents each trajectory of one space debris object. (See also the color figure section starting on p. 645.)

Fig. 3 Real-time trajectory estimation program. The green line represents the predicted azimuth/elevation/range. Once the real-time trajectory estimation runs, a red line will be shown in the same fields. (See also the color figure section starting on p. 645.)

debris (REPS) was adopted at TKSC in 2005. It is still in an early phase of experimentation; we implement its operation experimentally on some occasions. We plan to run the real-time reentry prediction for the last-hour space debris with this system. Figure 3 shows an example of the screen image of the REPS, and Fig. 4 represents the screen image of the reentry prediction supported by REPS. We can see the errors compared with the predicted orbit drawn by green in real time, and also see the real-time position drawn by a red line, where the tracking space debris is passing (see the color section of figures).

B. Optical Telescopes at BSGC

In observing the geostationary objects, we mainly use 1-m and 50-cm telescopes at almost an equal rate. Both are Cassegrain telescopes with equatorial mounts. The 1-m telescope can rotate at a velocity of more than 2.5 deg/s in both right ascension and celestial declination, and the 50-cm telescope can rotate at more than 5 deg/s in both directions. The 1-m telescope equips a cryogenically cooled CCD camera consisting of 10 2×4 k pixel CCD mosaics, whereas the 50-cm telescope equips a CCD camera consisting of a 2×4 k pixel CCD. The BSGC system is capable of detecting 16.5 to 18 magnitudes for both 1-m and 50-cm telescopes [4].

III. Observing Method of Space Debris

The KSGC radar is designed to observe mainly low-Earth orbit (LEO) objects, which are at about 400–1000 km altitude, and the BSGC 1-m and 50-cm telescopes

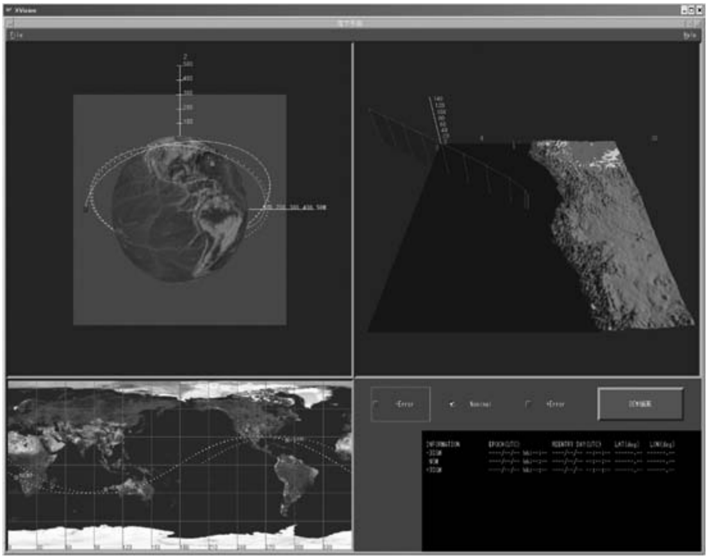


Fig. 4 Screen image of reentry prediction. The predicted reentry area (nominal and ± 20% prediction errors are included) is shown in the two-dimensional and three-dimensional world map. (See also the color figure section starting on p. 645.)

are mainly used to observe geostationary Earth-orbit (GEO) objects, which are at about 36000 km altitude.

A. Radar Observation

As for the method to track space debris, three modes are possessed: one to track them by using their initial orbital prediction to all the observable passes, another to track them by self-velocity prediction system, and the other to keep emitting pulses to the same direction to monitor the density at each altitude. To monitor rather large space debris objects passing over Japan and to monitor the behavior of the space debris generally, we are trying to observe as many space debris objects as we can at KSGC. When we choose the space debris to observe, the reentering objects are our first priority, and we narrow the targets down from all the other space debris to those for which the slant range should be less than 1350 km. After these objects are selected, we send the observation plan to the KSGC site. The KSGC radar can acquire the target space debris with ±20 s prediction errors. Once the space debris objects are primarily acquired, the radar keeps on tracking them. At TKSC we determine their orbits on weekdays automatically by using the three data observed during the continuous five days. We show the results of these observations in the next section.

B. Optical Observation

When people usually observe the astronomical objects, they set the telescope to the target objects and follow them across the sky as the Earth rotates. On the other

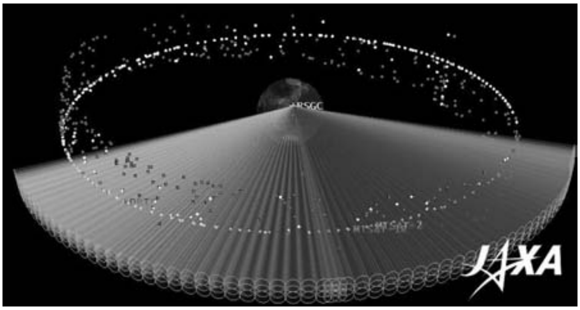


Fig. 5 JAXA GEO-belt survey. Yellow dots are operational satellites and red dots are space debris. One pink-colored circle represents a field of view for the 1-m telescope. The survey fields are separated into 120 deg. (See also the color figure section starting on p. 645.)

hand, in observing space debris at JAXA, we fix the telescope so that the astronomical objects appear to flow in one shot so that we can recognize the artificial objects easily since they move to different directions from the one shown by the stars. We distinguish that the objects found by this method are space debris for sure by taking some shots during a short period of time and detecting the same objects. Figure 5 shows the JAXA GEO survey areas. As this illustration shows, we mainly observe around the GEO-belt. First, we divide the belt into three rows, the upper, middle, and lower row. We start with the upper row and move on to the middle and the lower. Then, once we finish observing one column twice, we move on to the next column and keep on observing from one edge to another as long as we have visibility. At TKSC we determine their orbits on weekdays by using as many data as we could observe during the continuous five days. When the orbital elements are getting old, we specially plan the observation for them and maintenance the orbital elements. We show the result of these observations in the next section.

IV. Observation Results

A. Number of Observed or Catalogued Objects

As mentioned, we observe LEO objects by radar and GEO objects by the optical telescopes. Radar observation began in 2004 and optical observation began in 2000. Figure 6 shows our observation results.

In regard to the optical observation, we have paid attention to Japanese GEO objects for the first several years and spent most of the time observing them. Since late 2005 we have stepped forward to a larger area and started to observe the main GEO-belt. As of 31 May 2006, we nearly finished observing the objects twice whose longitude are from 65 to 185 deg east. That is, one survey is done in about half a year on average. If it takes too much time to finish one survey, we use two telescopes and can maintenance the orbital elements. As can be seen in Fig. 7, the number of observed or cataloged (= orbit determined) space objects is increasing

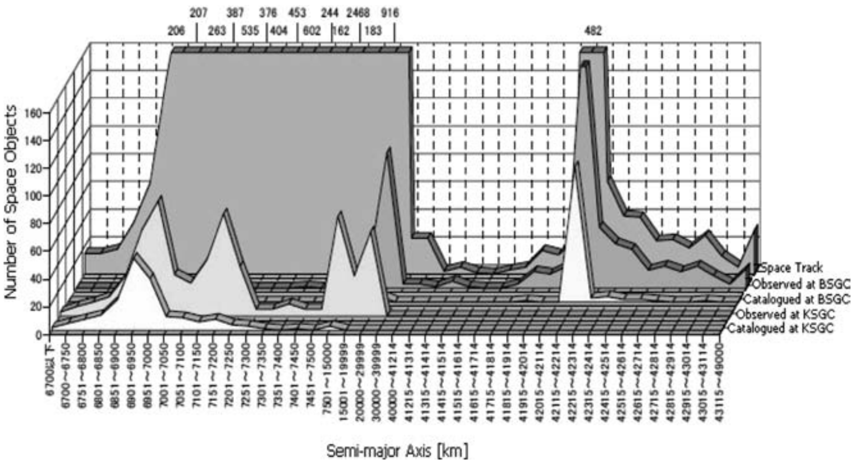


Fig. 6 Distribution of observed/cataloged objects (as of 31 May 2006): the yellow block, objects cataloged at KSGC; sky blue block, observed at KSGC; aqua block, cataloged at BSGC; pink block, observed at BSGC; and gray block, catalogued by Space Track. (See also the color figure section starting on p. 645.)

every month. As a result, referring to Fig. 5, we could observe both yellow dotted operational objects in the survey area and the red dotted non-operational objects, since non-operational objects are drifting around the Earth at several degrees per day, and we have the opportunity to observe them when they pass through our survey area (see the color section of figures).

In regard to the radar observation, compared with the optical observation, we do not have to care about the weather in this case, although the radar power is

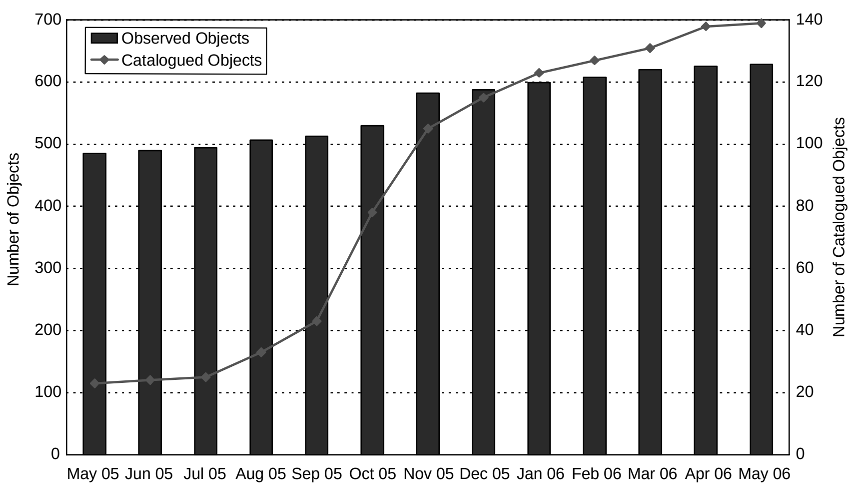


Fig. 7 Number of observed/cataloged objects at BSGC.

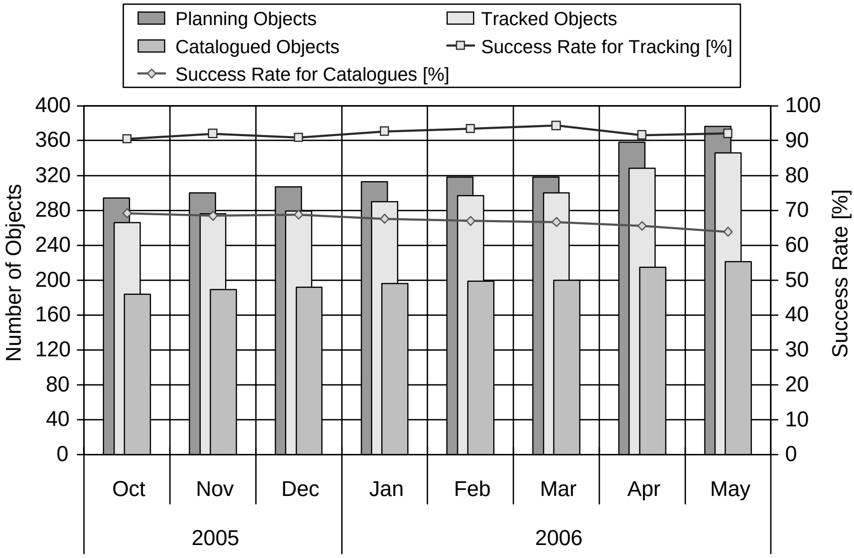


Fig. 8 Number of observed/cataloged objects at KSGC.

limited. To maximize the radar capability, we devise the most suitable parameters to each characteristic space debris object on a regular basis, and these efforts can be seen by the increase in the number of planning objects we are trying to observe, as shown in Fig. 8.

B. Accuracy of Orbit Determination

To evaluate the accuracy of our orbital determination, we compared the orbital elements that are determined by the data taken one day before and that taken on that day, by propagating the one-day-before orbital elements for one day, so that they should be on the same epoch each other.

In regard to the radar observation, the mean errors of all the space debris taken by radar are in the semimajor axis $\Delta a \cong 22.0$ m, in position $\Delta R \cong 5.57$ km, and in inclination $\Delta i \cong 0.00852$ deg. If we take only one space debris object, ETS-VII, for example, which we ended its operation in 2002, we had errors in the semimajor axis $\Delta a \cong 8.35$ m, in position $\Delta R \cong 2.26$ km, and in inclination $\Delta i \cong 0.00380$ deg. Figure 9 shows the errors of the orbital determination of ETS-VII from 2004 through 2005.

On the other hand, with regard to the optical observation, the mean errors of the orbit determination are in the semimajor axis $\Delta a \cong 67.7$ m, in position $\Delta R \cong 15.7$ km as of 31 May 2006.

As a conclusion on orbit determination, we can say that our data are satisfactory from the viewpoint that we can ensure the recursive observation against LEO and GEO space debris, even if we allow the ambiguity of approximately 25 m by radar caused by signal procession, noises of data, and the constraints of the jump of

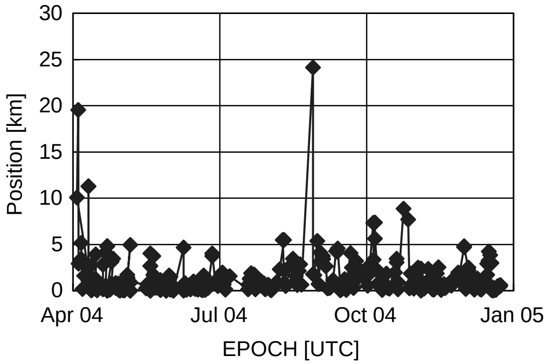


Fig. 9 ΔR of ETS-VII orbit determination, comparing the difference in position by propagating the one-day-before determined orbital elements for one day.

radar cross section (RCS), which would happen since the attitude of space debris changes rapidly as they are non-operational objects.

C. Reentry Prediction

In regard to the space debris that are to reenter the atmosphere in the near future, we regularly track the target, determine its orbit, and predict the time of reentry, as needed. Since we started the observation of COSMOS 2332 as a first try in 2004, we are keeping our eyes on all the reentering objects observable at KSGC. Figure 10 shows one of the major reentry examples we predicted last year.

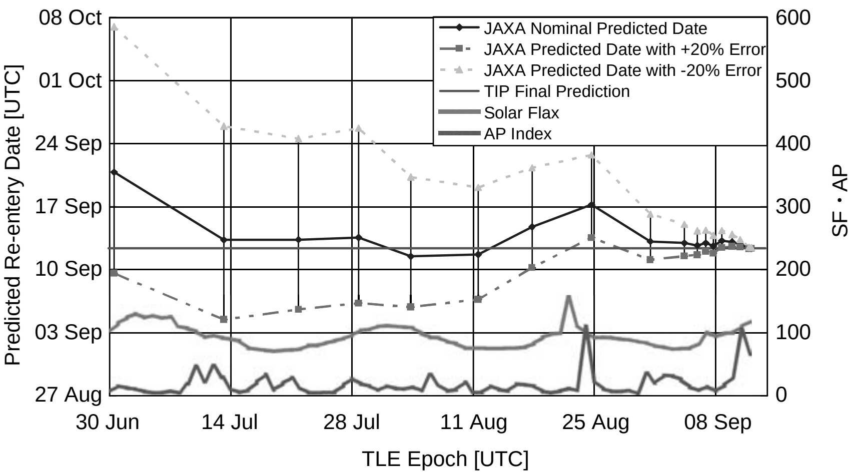


Fig. 10 Predicted date of YOHKOH's reentry. (See also the color figure section starting on p. 645.)

We started to predict the decay of YOHKOH, a Japanese satellite, two months before its decay and ran the calculation every week. Once the predicted date became one week before its reentry, we ran the prediction every day. We've learned the errors of reentry prediction caused by the activity of the sun are estimated to ±20% statistically by the past prediction.

V. Plan for Future Operations

To plan for the next few years of radar observation, the central aim is to observe all of the approximately 300 space debris objects, which should be observable at KSGC in full time according to their orbits. We are attempting to improve the detection capability and to observe space debris as much as possible with this radar. To obtain more capability, we try to assess the system for capturing and tracking space debris, and to improve our signal processing to save the small signals buried in the noise. So far, the average values of our analyzed RCS are showing the slightly higher values than those of the U.S. Space Track. In our system, we put these RCS values to choose that distance to wait, which depends on how large the RCS of the target objects are. Hence it is very important to evaluate the RCS value for each object accurately. On the other hand, our RCS values sometimes work very well for some objects and sometimes they do not. Although many people know this is a very hard topic, we also find a lot of difficulties in evaluating the appropriate RCS value of each object, for most objects are spinning randomly and possess complicated configurations. However, by having the appropriate RCS values, we are aiming to track more objects in the future.

For the optical observation, the GEO-belt survey is working very well so far, but in order to obtain more cataloged objects, we are considering starting to communicate with other telescopes in Japan. By communicating with each other, we are aiming to observe more efficiently together. Another thing we are trying is that we would like to determine the orbits using fewer data taken over 10 days or so. Because the weather conditions are bad in Japan from spring through fall for optical observation, this is also a critical challenge for us.

VI. Conclusion

Three years have passed since the KSGC radar started the operation and seven years have passed since the BSGC telescopes started observation. In this process, we cataloged 221 LEO objects and 139 GEO/GTO objects as of 31 May 2006. Because the power of the radar is limited and optical telescopes depend on weather deeply, it is not easy to improve the number of cataloged objects. Still, we would like to try even little devices and would like to improve these numbers as written in the former section. In regard to the cooperation between the radar observation and the reentry prediction, we have obtained the simple method. Hence further challenges are to observe the reentering objects and to make decay predictions using our data more precisely. On the other hand, in RCS we still have some jumps or lack of the data, considered to be originated to the spinning motion of space debris, and need to consider how to solve these difficulties under operations. In the near future it will be necessary to give more feedback to our observation by careful evaluation of our observed data.

Acknowledgments

The authors thank the observers/operators of BSGC and KSGC.

References

[1] *US Space Track Homepage* [online database], http://www.space-track.org/ [retrieved 31 March 2006].

[2] Ono, K., et al., "Development of the First Japanese Space Debris Observation Radar," *Third European Conference on Space Debris*, Vol. 1, 2001, pp. 73–78.

[3] Sawabe, M., et al., "Overview of NASDA Orbit Analysis System for Space Debris and Bisei/Kamisaibara Spaceguard Centers," *ISTS*, 2002-r-14, 2002.

[4] Okumura, S., Isobe, S., and JSGA team, "Systematic Survey Observations of Space Debris During These 5 Years," *IAC*, IAC-05-B6.1.06, 2005.

This page intentionally left blank

V. Automation

This page intentionally left blank

Chapter 18

Automation of ESOC Mission Operations

G. P. Calzolari,* S. Haag,† C. R. Haddow,‡ and M. Pecchioli§
European Space Agency, Darmstadt, Germany

I. Introduction

THE European Space Operations Centre (ESOC) has recently finalized its concept for the increased automation of future ground segment and spacecraft operations. To utilize the experience gained in the past and ensure a smooth transition toward operations automation, the adopted concept mirrors the currently used operational model. This splits responsibility for mission operations typically into two parts. The first of these deals with the allocation and operations of shared resources, i.e., ground stations, communications links, etc., the second with the actual operations of the spacecraft and associated mission-dedicated resources as well as the usage of allocated shared resources. This operational model has proved to be successful and robust over many missions as it is based on a clear distinction of responsibility. In view of this, it has been decided to maintain this model in the agreed approach to increasing the automation of segment operations. Consequently, two new systems are being developed to provide the functionality needed to increase automation in the two areas of responsibility.

The first of these is the ESTRACK management and scheduling system (EMS). The ESTRACK management and scheduling system is a suite of applications that support the automated planning and scheduling and the centralized coordination of ESA's ESTRACK network. The ESTRACK network comprises all of the ESA facilities deployed around the world to provide the tracking services required by the agency and its customers. This includes ground stations, communications, and

---

*System Engineer, Ground Station Backend Section, European Space Operations Centre, OPS-GIB.

†Software Engineer, Operation Centre System Infrastructure Section, European Space Operations Centre, OPS-GIC.

‡Senior Software Engineer, Data System Infrastructure Division, European Space Operations Centre, OPS-GI.

§Head of the Operation Centre System Infrastructure Section, European Space Operations Centre, OPS-GIC.

control facilities. The ESTRACK network is now large and complex, and it keeps growing. Stations are remotely operated on a routine basis. They are supporting multiple missions, within and outside ESA. The requests from the users are evolving from direct request of specific facilities to more generic tracking service requirements. These new needs have driven the requirements for an automated planning and scheduling and global network monitoring and control system, the functionality of which is provided by the EMS.

To support the automation of mission dedicated parts of the ground segment and spacecraft operation, a second system is being developed, the mission automation system (MATIS). This will be driven by schedules generated by a particular mission's mission planning system (MPS) and will control the operations of the mission control and network interface systems. MATIS will therefore be responsible for the initiation of connections to the ground station, the uplink of commands to the spacecraft, the starting of specific processing activities, etc.

Access from MATIS to the required services supplied by the different control data systems will be achieved through a standardized interface layer. This standardized layer is called the service management framework (SMF) and plays a key role in the approach to automation as it is intended to ensure complete transparency of the actual requests implementation details, e.g., current location of the process supporting a given service and the exact details of how the underlying service is provided. In essence the SMF can be considered an encapsulation layer for the control system data systems.

II. System Context

A typical ESA ground segment is composed of a number of individual elements, some dedicated to a particular mission, others shared between different missions. Figure 1 illustrates the system context within which the automation concept operates. The main systems involved in the ground segment are discussed in the following sections.

A. Shared Resources

The shared resources in the ESA ground segment are as follows:

1) Flight dynamics systems (FDS). This system supports the determination, monitoring, prediction, and active control of the spacecraft orbit and attitude. The role that is played by the FDS during both the planning phase and the execution phase strictly depends on the mission objectives and design.

2) Station computer (STC), one per ground station. This computer supports the remote monitoring and control of ground stations equipment, based on the automatic execution of the schedules received from the EMS. Currently the role of the STC at ESTRACK is covered by two different implementations, the STC-2 and the centralized station monitoring and control (CSMC).

3) Ground station (GS) equipment, used by several missions on a time-shared basis. Utilization of GS equipment depends on mission specification.

4) EMS ESTRACK management system. This system is discussed in more detail in Sec. III.

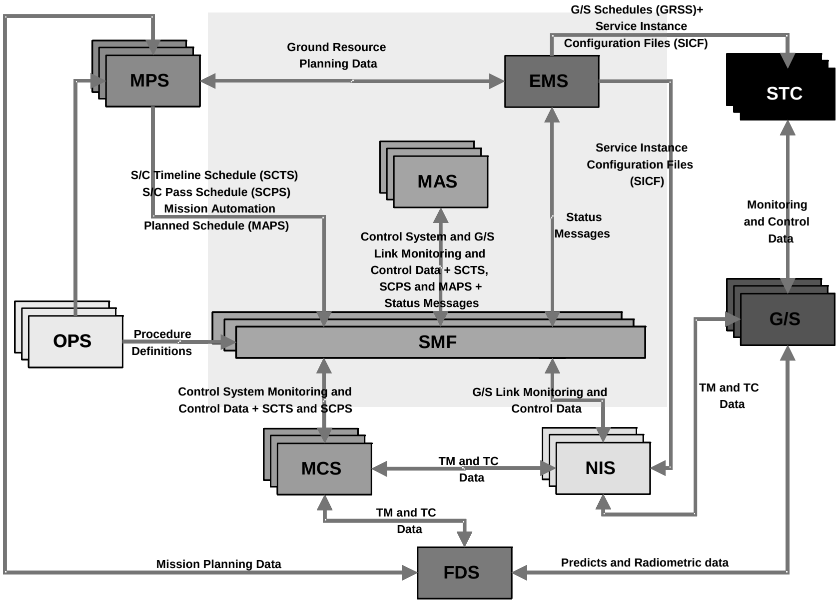


Fig. 1 System context within which the automation concept operates. Shaded area contains those systems providing the main elements required to enable automation.

B. Mission-Dedicated Resources

The mission-dedicated resources in the ESA ground segment are as follows:

1) Mission planning system (MPS). This system is responsible for generating the initial inputs to the EMS and using the resultant schedule to generate a sequence of operations for the spacecraft and control system. The MPS could in principal also be an entity external to ESOC.

2) Operations preparation system (OPS). This system is used by the flight control team (FCT) in the preparation of procedures for operating the spacecraft.

3) Mission control system (MCS). This provides the main facilities for monitoring and controlling the spacecraft, e.g., telemetry and telecommanding, etc.

4) Network interface system (NIS). This system is responsible for controlling the links between the MCS and the individual ground stations. All interactions between the NIS and the ground stations use the Consultative Committee for Space Data Systems (CCSDS) space link extension (SLE) [1] protocols.

5) Mission automation system (MATIS). This system is discussed in more detail in Sec. IV.

6) Service management framework (SMF). This is discussed in more detail in Sec. V.

III. ESTRACK Management System

The ESTRACK management and scheduling system (EMS) is a suite of applications that support the automated planning and scheduling, and the centralized

coordination of ESA's ESTRACK network. The ESTRACK network comprises all of the ESA facilities deployed around the world to provide the tracking services required by the agency. This includes ground stations, communications, and control facilities.

Historically ground stations were almost exclusively dedicated to a given ESA mission, operated locally by the ground station staff with practically no systems between the ground stations and the control system (only communications lines when the operations center was co-located on the ground station site). The network management and scheduling systems were designed in accordance with those principles, and planning performed manually at the scheduling office. Ground stations were scheduled as directly requested by the mission as exclusive users of the facilities.

The ESTRACK network has now grown in size, capability, and also in complexity, and it is still growing. Stations are remotely operated on a routine basis. They are supporting multiple missions, within and outside ESA. The requests from the users are evolving from direct request of specific facilities to more generic tracking service requirements. These new requirements have led to the development of a new network management and scheduling operational concept, based on automated planning and scheduling and global network monitoring and control, which is supported by EMS.

An additional item needing serious consideration by EMS is the increase of interoperability issues (e.g., the usage of CCSDS Space Link Extension (SLE) services) that allows external users to access ESTRACK as well as allowing ESA missions to obtain services provided by external networks and ground stations. This has been taken into account in the EMS by allowing exchange of information with external entities via the concept of a dedicated proxy for each type of external provider. A proxy for the NASA Deep Space Network (DSN) is part of the EMS baseline.

EMS itself comprises three major components (as shown in Fig. 2), which are loosely coupled and can be operated independently to a large extent: the ESTRACK planning system (EPS); the ESTRACK scheduling system (ESS), which coverts the allocation plans generated by EPS in schedules that can be executed automatically or manually at the ESTRACK facilities; and the ESTRACK coordination system (ECS), which ensures the coordination of the network service allocation at run time.

A. ESTRACK Planning System

The EPS is organized around a plan generator, which is responsible for the allocation of the ESTRACK services to the mission on the ESTRACK management plan (EMP). This plan generator relies, mainly, on two types of information to maintain the ESTRACK management plan:

1) Static data kept in the EPS configuration database. This covers models for the mission requirements for service allocation, represented by mission agreements, and models of the services that are provided by the various ESTRACK facilities.

2) Dynamic data provided routinely by the mission during the operational phase. This covers event files including predictions for the events relevant to the service allocation for the mission, dynamic refinements to the mission

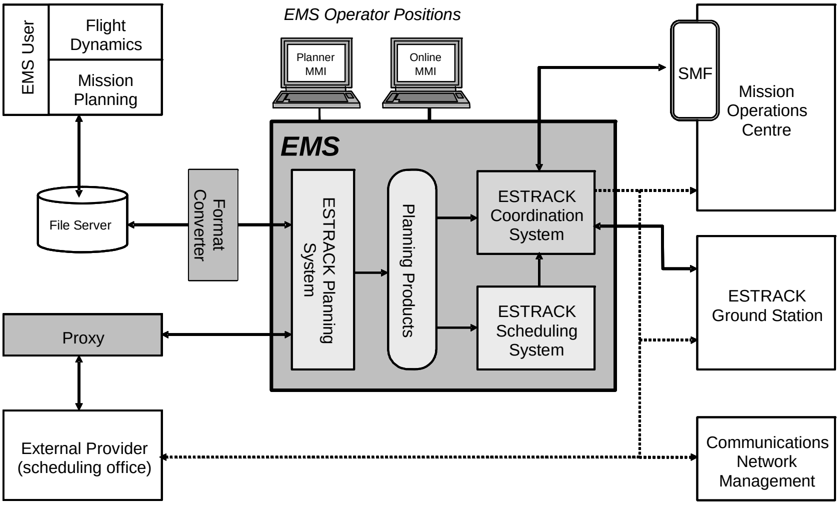


Fig. 2 EMS operational context. Shaded area contains the subsystems that together comprise the EMS.

requirements of the mission agreements, and refinements to the service sessions allocated to the mission themselves.

In addition, the plan generator has to take into account allocation plans received from external providers. By these means the EPS can also support the planning of operational services provided by external networks. Because ESA supports a wide variety of mission types, the planning cycle needs to be flexible to accommodate the specific planning cycle of the different missions. The cycle of interaction between a mission and EPS for the planning of the service allocation at a time *T* is as follows:

1) The mission provides EPS with event files including the orbital and mission event predictions.

2) The mission provides EPS with optional order refinements.

3) EPS updates the ESTRACK Management Plan (EMP) and generates mission-specific plan views.

4) The mission retrieves the plan views.

5) The mission commits selected service sessions if the earliest commit time is reached (optional) and before the latest commit time is reached (mandatory).

6) The mission refines selected services sessions (optional).

7) If the scheduling time is reached, EPS generates plan views needed for scheduling tool; otherwise go to step 1.

This cycle reflects the interaction that can take place during the medium-term planning of the network service allocation. To take into account long-term predictions, and also to be able to accommodate late requests and modification to the EMP, the global planning cycle is separated into the three typical phases of long-term, medium-term, and short-term planning.

The long-term planning starts as soon as all orbital predictions are available for a time *T* for all missions supported by EPS. At this stage the EPS planner can run

the system to derive long-term allocation plans, but it is not foreseen that the mission will be given the corresponding plan views as input, as they anyway lack the accuracy needed for the detailed mission planning.

The medium-term planning phase is foreseen for routine service allocation. During this phase, additional passes (non-routine contact) can be requested via order refinement, but will be rejected by EPS if they prevent the allocation of the routine passes (i.e., defined by the nominal planning directives) to all missions.

The routine service allocation is performed based on the planning directives of the mission agreements, with the standing orders affected by the accepted order refinements. The planning directives are qualified to represent three levels of service for each mission: optimal, nominal, or degraded. The routine service allocation ensures at least the nominal service level (i.e., nominal planning directives) is implemented. It will provide the optimal level of service if possible. In case of overloading of the network (for instance, if one or more stations become unavailable at the same time), the operator can activate the degraded planning directives for some missions, therefore allowing for a valid EMP to be generated, at the cost of reducing the level of service for these missions. All of the activated standing orders are taken into account during planning. The de-activated standing orders can be activated if they are called within an order refinement.

The short-term planning starts after the phase of routine service allocation and concentrates on distributing the network resources still available. In this phase, it is not expected that any significant update to the event predictions is input to the system. All passes committed at the end of the medium-term planning are protected during the short-term planning service allocation. It is only in case of emergency that already committed passes can be reassigned to missions by the EPS operator.

B. ESTRACK Scheduling System

The ESTRACK scheduling system (ESS) is the component of the EMS that translates the abstract plans generated by the EPS into executable schedules. Schedules are produced for all ESTRACK facilities, including the ground stations and the data communications network. A schedule can be produced to be directly executable by a ground station or to be a list of operations to be performed by the station operators.

In addition, the ESS generates a master schedule for the ESTRACK coordination system (ECS), which is used to coordinate the download and execution of the schedules created for the other facilities. The master schedule drives the ESTRACK coordination system, when this system is operating in automated mode.

The ESS is also responsible for the generation of the service instance configuration files (SICFs), for both SLE (SLE-SICF) and file transfer (FT-SICF) transfer services, for distribution to the mission operations center, ground stations, and optionally to service providers.

When ESS generates executable schedules, it additionally generates a human readable time line for the operations team. The master schedule is the facility schedule for the ESTRACK coordination system. It implements the downloading of schedules to ground stations and the network management system, the commands to start schedules on ground stations and the invocation of coordination procedures

when these are supported. The scheduler also checks that constraints imposed by each facility on the minimum lead time when a schedule has to be present at the facility before the planned start of schedule execution are met.

C. ESTRACK Coordination System

The ESTRACK coordination system (ECS) is the EMS component that implements the online management features of EMS. ECS receives the frozen portion of the ESTRACK managements plan from the ESTRACK planning system and the operational schedules from the ESTRACK scheduling system and provides the following main features: 1) downloading of schedules generated by the ESTRACK scheduling system; 2) monitoring of service provisioning and schedule execution; 3) schedule control; 4) coordination of facilities, and possibly external providers, for special procedures; 5) logging of all events and generation of session reports; and 6) execution of the EMS master schedule provided by the ESTRACK scheduling system.

Downloading of schedules will be performed at a specified time before the schedule start time and during periods where the target facility is not supporting an online service session.

Monitoring of schedule processing and service provisioning will provide a synoptic view of planning information and status information retrieved from ground stations and (optionally) the mission control data systems. The monitor information presented by ECS shall allow the operator to assess the status of schedule processing and of the facility and shall not include direct equipment monitoring. In case of problems, an alarm shall be raised and the operators will then use the ground station management system for problem analysis and correction.

Schedule control will enable the EMS operator to start and stop schedules on the facilities. In addition, the EMS operator will receive a message before a schedule executes a critical operation and needs to confirm or reject the operation.

To monitor the status of the mission control data systems, ECS needs to exchange information with these systems. For this purpose, ECS will use the SMF services provided by the mission-dedicated systems. To maximize decoupling and minimize dependencies, the EMS will be able to operate without reliance on any data exchanged through this interface.

The ECS will be operated by the team responsible for operation of the ESA ground stations with the operator position being located in the ground facilities control center (GFCC).

IV. Mission Automation System

The mission automation system (MATIS) provides the mission-dedicated system controlling the automatic operations of the mission-dedicated systems. Two major distinct operational environments are foreseen in the MATIS operational concept: 1) a preparation environment, dedicated to schedule preparation and maintenance; and 2) an execution environment, dedicated to schedule and procedures execution.

There is a clear distinction between the preparation and execution environments. The preparation environment provides offline functionality, allowing the user to prepare and validate mission automation users schedules (MAUS) within

a provided authoring tool (MAUS editor), while the execution environment allows execution of MAUSs, mission automation planned schedules (MAPS) (i.e., the input from the mission planning system), as well execution of stand-alone procedures not referenced by any schedule. The execution environment layering is illustrated in Fig. 3. Within this environment the execution of more than one schedule in parallel is permitted, each with its own timing constraints and reference to specific procedures.

MAPS and MAUS are defined as sets of tasks (e.g., procedures) to be executed within the schedule timeline within sequential or parallel branches. Interlocks among tasks exist in terms of time and dependencies constraints, and the two allow the user to include in the schedule logic control of the execution flow of tasks. The set of tasks and branches is defined as schedule layout.

At schedule definition level a task is an entity to be executed and can be a procedure, a reference to an event (predicted or external), or a checkpoint (used for replanning). While it is clear that the main objective of schedules is to trigger execution of procedures (in parallel or sequential order according to the schedule layout), they are not the only entity a schedule can define. The support for events and checkpoints are part of the schedule definition in order to model the following:

1) The need to define predicted events at schedule level. A predicted event is a known event a schedule is required to reference to start the execution of a task.

2) The need to wait for an external event. This can be used to trigger the execution of a branch based on the occurrence of the external event

3) The need to define clear schedule execution status “milestones” (checkpoints) used to synchronize the execution of a schedule with another schedule or to replan it with another version of the same schedule (replanning). Checkpoints are defined by the user (e.g., MPS for MAPS), which when reached during the execution of a

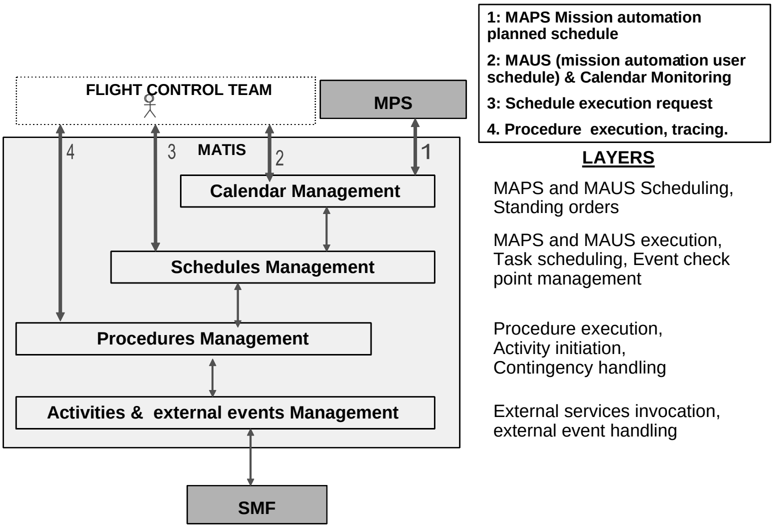


Fig. 3 MATIS execution layers.

schedule, identify a known status of the overall execution of a schedule so that operations such as replanning (or handover to another schedule) can be performed with a clear understanding of the execution status.

Both types of schedule do not include the definition of procedures but only make reference to them. This means that, at system level, MATIS relies on a concept based on the following:

1) Procedures are prepared and maintained outside the system [in the Operation Preparation System (OPS)] and are made available to MATIS as needed. In case procedures maintenance is required, OPS is responsible for providing the new version of all applicable procedures and associated telemetry and telecommand (TM/TC) definitions to MATIS. In other words, procedure and TM/TC configuration control is performed by OPS and MATIS needs only to have a copy of the latest version. Alternatively, the MATIS Mission Database (MDB) could provide version control for procedures, but the mission can determine this.

2) MAPS is produced by MPS.

3) MAUS is produced by FCT using the MATIS preparation environment. While preparing MAUS, FCT will need to have visibility of the definition of the currently available procedures to prepare a consistent MAUS, that is a MAUS that includes references to procedures (including their signature in terms of number of input parameters and their type, etc.) available in the execution environment.

4) Because MAPS are prepared by MPS, it is clear that MATIS and the MPS must be aligned in terms of a valid set of procedures that can be referenced by a schedule (either MAPS or MAUS) at any one time. In other words, whenever a procedure or TM/TC definition maintenance session is performed, OPS is required to submit the new set of valid procedures and TM/TC definitions to both MPS and MATIS as needed.

MATIS procedures will be defined in the PLUTO [2] language and shall be made available by OPS in terms of source code files together with their definition within a procedure definition file (PRDF). The procedure definition file simply describes the procedure (description, number, and type of input parameters, etc.) together with a procedure reference name. The procedure reference name is used by the MAPS and MAUS as a task reference to a procedure. The usage of such alias allows the decoupling of the reference to a procedure to its actual implementation, thus enabling maintenance of the procedures independently to their MAPS and MAUS definitions.

Procedures make reference to activities (external services) or to other procedures. Activities are services offered by the external world (via SMF) or by MATIS itself (the invocation of procedures from within procedures will be a special MATIS service) and are used by a procedure to interact with the external world to fulfil the objectives of the procedure main body.

MATIS will control the operation of mission-dedicated systems by means of services exposed through SMF. All services available from within a MATIS procedure will be configured in a service definition file (SRDF). The service definitions file shall be stored in the MDB as persistent configuration data for a particular mission. The SRDF defines for each service its name and required parameters and allows a configurable way for a mission to define which services may be referenced from with a procedure. The service definition file will also need to be made available to OPS for consistency checking and validation of

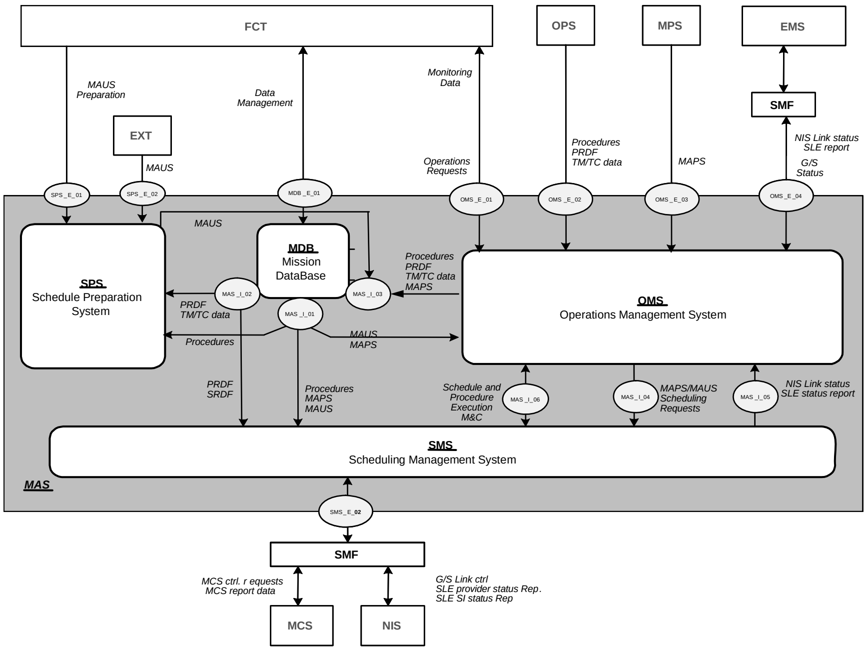


Fig. 4 MATIS operational context. Shaded area contains the subsystems that together comprise MATIS.

procedures. Contingency handling at schedule and procedure level will also be supported: 1) at schedule level with the support of events and branches, it is possible to execute a number of tasks according to the occurrence of an event, and 2) at procedure level it is possible to execute a number of tasks using the PLUTO watchdog body.

Figure 4 illustrates the system context for MATIS, along with its main constituent subsystems. These are described further in the following sections.

A. Schedule Preparation System (SPS)

The MATIS schedule preparation system (SPS) is an offline preparation environment of MATIS, where the FCT can develop and validate MAUS with the support of the procedure definitions prepared by OPS and stored in MATIS MDB. It will provide a MAUS schedule editor, which allows the user to create new MAUS schedules or edit existing MAUS schedules.

In addition, the user will be able to validate a created/edited MAUS against the current set of MATIS procedures, events, and checkpoints. It shall also be possible to load a MAUS received from an external entity into the MAUS schedule editor and perform editing and validation. The use of MAUS is not imposed by the concept nor by the implementation of MATIS. It is an additional mechanism to allow the production of schedules, which complements the ones produced by MPS

(if needed), i.e., MATIS may be used solely with MAPS. The MAUS is primarily intended to allow the FCT to add ad hoc procedures without the need to incorporate these into the mission planning system.

The MATIS SPS will support privilege control for access to the different editing and validation functions. It shall furthermore be possible to have any number of configured users running a MATIS MAUS schedule editor at the same time.

B. Scheduling Management System

The scheduling management system (SMS) is responsible for the actual real-time execution of the schedules authorized in MATIS. Multiple schedules may be running at a time, and schedules may contain parallel executing activities. The SMS is in charge of keeping track of all of the tasks associated to activities referenced by running schedules and to ensure the associated activities (i.e., procedures) are executed at the specified times maintaining any defined execution constraints, etc.

C. Operations Management System

The operations management system (OMS) is responsible for implementing the operational management logic of MATIS at system level. It is the OMS that handles requests for execution of a MAUS and/or MAPS, as well as handling interfaces to OPS, MPS, and EMS. The level of automation included in any instance of MATIS is implemented at system level by OMS in the way requests of operations are handled.

In single- and multidomain environments, the system mission context could include just one instance of MATIS and the level of automatic operations and coordination among the different interfaces (OPS, MPS, FCT, and EMS) and related requests are mapped in MATIS within its OMS logic.

To fulfill the MATIS assigned operations logic, the OMS makes use of SMS services. Once an entity (schedule, task) is submitted to SMS, it is processed as instructed, that is as per the logic included in the schedules and procedures. SMS is, however, always triggered by the OMS issuing an SMS service request and SMS always refers to OMS for decision actions.

D. Mission Database

To perform the required activities, the MATIS includes a MBD system, which is responsible for maintaining static data definitions and configuration data, e.g., activity definitions (procedures), TM/TC definitions, etc. The MDB also provides a version control mechanism for stored data including procedures, schedules, and configuration data.

V. Service Management Framework

The service management framework (SMF) is a generic service interface middleware providing a set of low-level mechanisms capable of exposing services

according to the service-independent interface specifications. Its main capabilities are the following: 1) definition of a standard and general mechanism to expose services, 2) independence of the framework from the systems providing the services, 3) scalability and flexibility of the architecture and the run-time environment, 4) transparency for the service consumers regarding the implementation and deployment details of the service providers, 5) capability to distribute the framework tasks on the network, and 6) capability to allow access to exposed services only to authorized users.

Figure 5 illustrates the system context of the SMF and identifies its main components. It should be noted that in this context external user refers to any entity that utilizes the services exposed by the SMF whether it is a human user or other computer system.

The service providers, i.e., the applications on the encapsulated systems that are responsible for providing the required services, are referred to as application units (AU). Such applications can be already existing applications (such as provided by the existing mission control system applications), able to provide the required services [referred to as legacy application units (LAU)], and/or specific applications that need to be developed to provide particular services. Internal to the SMF the various components are also structured as application units, so that the mechanisms used by the SMF to internally access services are the same as those used to access external services.

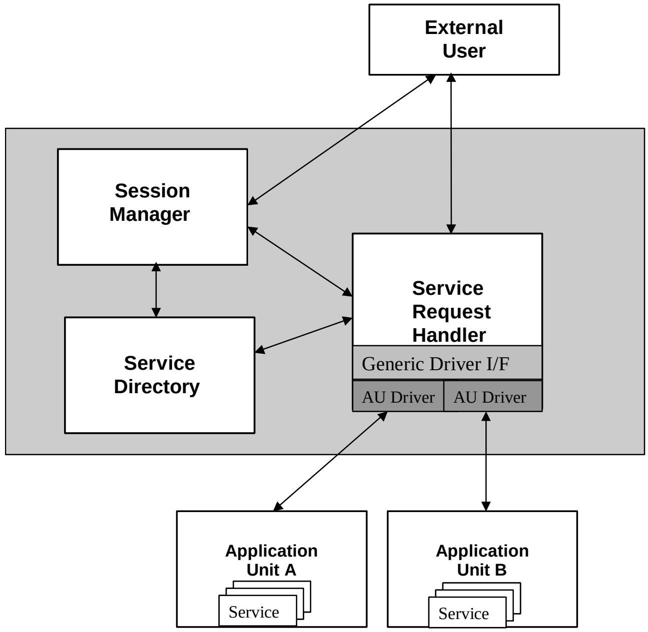


Fig. 5 SMF operational context. Shaded area contains the components that together comprise the SMF.

In designing the SMF considerable thought was given to producing an architecture that would be expandable to cover most of the systems in a ground segment. Thus, while the initial delivery of the SMF will only support services provided by the MCS and NIS, in future it is envisaged that this could be expanded to cover the FDS and MPS systems.

SMF exposes services according to ECSS-70-31 Space System Model [3]. All services are described in XML files as a tree of system elements. Each system element is composed of: 1) activities, to initiate actions on the system; 2) reporting data, to get/set the data describing the status of the system; and 3) events, to notify external users of system changes.

A. Session Manager

The session manager is an internal SMF application unit that is responsible for the management of access to the services from external users and is the access point to issue a reconfiguration service execution that has the effect to reconfigure SMF for serving services of a new server family. Only authorized external users are permitted access to the services exposed by the SMF. For this reason the external user has to open an SMF session. Each user is associated with a user profile describing which services they can have access to.

Each session has an associated operative mode that limits the way the session owner can access the exposed services. The possible operative modes are as follows:

1) Monitoring mode. In this mode it is possible to access all of the services that do not change the status of the system.

2) Monitoring and command mode. In this mode it is possible to have access to all of the services related to the operational usage of the SMF (e.g., command injecting) plus all of the services possible in monitor mode.

3) Monitor and control mode. In this mode it is possible to access all of the services related to the configuration of the SMF and the system controlled through the SMF (e.g., start/stop/monitor task, initiate reconfiguration) plus all of the services possible in monitor mode.

4) Administrator mode. In this access mode it is possible to access all of the available services.

B. Service Directory

The SMF is a distributed system and requires a central point to locate the exposed services and the application unit that exposes them. For this reason there is a service directory that contains such information. The mechanism implementing the service request makes access to the service directory transparent to the user. The main role of the service directory is to contain all of the relevant data necessary to identify the location of the system elements exposed by the service management framework. It also contains a static description of the possible services (system element) that may be exposed via the SMF.

The service directory system exposes services to: 1) register services and application unit location, 2) get the application unit location, 3) start SMF task applications, and 4) stop SMF task application.

Note that although not required by the implemented mechanism, an authorized user can directly request the location of the services from the service directory. The service directory is also used as central repository for the service description.

The service directory has also been implemented as an internal SMF application Unit utilizing OpenLDAP as the underlying repository.

C. Service Request Handler

The service request handler (SRH) represents the separation layer between the external user and the application unit that wants to expose automation services. It hides the application unit details to the external user. The service request handler allows the external user to see the system that exposes services as a set of system elements.

The scope of the service request handler is to accept the service requests performed by the external user. It does not perform any processing for the service requests; it verifies the privileges and roles and performs, if enabled, a first consistency check of the input parameter, i.e., check if the service request is correctly formed.

The SRH behaves as a router for the service requests. In case of any type of exception in the request (e.g., request formatting error or privilege failure), an error is reported to the external user. The real connection between the external user and the application units exposing the service is performed using the application unit drivers. Each service request handler instance is in charge of the management of a set of application unit drivers. The automation unit drivers are instantiated inside the service request handler during the startup. The driver offers a standard Application Programming Interface (API) layer that permits the execution of the external user service requests.

To provide additional flexibility, it is possible to configure the drivers such that they can run on the same physical host as the SMF or alternatively on the system hosting the application that is providing the service. Communications between SMF components are in general based on CORBA; however, between the driver and the application unit, CORBA or TCP/IP can be used, depending on what the application unit can best support.

VI. Conclusion

A. Status

The status of the various elements involved in the ESOC automation concept at the time of writing (March 2006) is as follows:

1) For EMS, development is following a phased approach with the EPS being the first part to be developed. Currently this is being implemented and the first delivery is scheduled for Q4/2006. The ESS is the next element to be delivered, and the implementation phase has just kicked off with the first delivery due Q3/2007. The final part of the EMS is the ECS, and it is currently planned that development of this will start Q4/2006 with the first delivery due in Q1/2008.

2) For MATIS, the architectural design has started and the first delivery is due Q2/2007.

3) For SMF, development of the initial version has been completed and is currently available.

In addition, there are updates required to existing systems to support the automation. The enhancements to the MCS systems are currently under way, and the set of drivers required to expose services via the SMF will be available in Release 5 of the S2K software. This is scheduled to be available Q4/2006. The final element needed in the initial phases on the automation, the NIS [4], is currently undergoing preliminary testing and is scheduled to be available Q4/2006.

B. Issues

There are still a number of outstanding issues that need to be addressed, such as the following:

1) No infrastructure is available in the medium-term in the area of mission planning systems. This implies that missions will have to develop their own interfaces between their mission planning systems and EMS and MATIS.

2) It is not yet clear when (if) other systems (e.g., FDS) will be enhanced to provide SMF support.

3) The issue of procedure definition and validation will need to be addressed. This is particularly true of contingency recovery procedures. Consider the case today where an anomaly occurs on the spacecraft. In such a case when the spacecraft controller observes an anomaly (for example, in the telemetry a parameter is reported as out of limits), they will follow a written procedure. This procedure will have a number of steps that have to be executed and may well have a number of branch points, depending on what the results of the various steps in the recovery procedure are. To encapsulate the steps in such a procedure in a way that can be executed by an automation system can be challenging and may require a change in the way in which the flight control teams prepare their procedures. Even more challenging will be the debugging of automated recovery procedures.

C. Final Objectives

From the foregoing discussion it can be seen that the concept that will be used in the automation of operations at ESOC has been finalized, and the implementation of almost all of the required components has begun with the remainder expected to be started in the coming months.

The objectives are clear: the ESTRACK management system will provide the means by which the scheduling of shared resources is carried out. This will take into account not only ESA missions but also the scheduling of ESA resources for external missions and will incorporate the allocation of resources from external agencies (e.g., NAScA's Deep Space Network) when producing the schedule of ETRACK resources. It will also provide the means by which these schedules are automatically executed and monitored. This will be a centralized, redundant system.

The mission automation system will provide the means by which control of mission dedicated systems are automated. Schedules used by the MATIS will be generated by each mission's own planning system. The planning systems will take into account the ESTRACK resource allocations generated by the EMS as well as

mission-specific constraints and observation requests. Unlike the EMS, MATIS will be mission specific, and there will thus be one MATIS per mission. It should be noted that a mission could consist of more than one spacecraft.

The service management framework provides the mechanism that enables control of the mission-dedicated systems to be automated. As it encapsulates the underlying systems and only exposes a set of standardized services, it hides the details of the internal workings of the mission-specific systems, thus making definitions of the schedules required to execute it more straightforward. Similar to MATIS, there will be one SMF per mission. The SMF also provides a decoupling between the systems. This is essential, as failure of one system must not be able to propagate into others. It should be noted that the SMF is not only applicable to the automation of operations but will also be used to provide services to users external to ESOC.

The concept developed is also flexible in that it does not mandate that a mission automate its operations. These could continue to be carried out manually using the output from the EMS in the same manner as the current paper output from the station scheduling office is used today.

Indeed it is unlikely that any mission will use the MATIS for “lights out” operations in the near future, as there are a number of issues that would preclude this. For example, the systems are complex, and no matter how much testing is carried out, it will take time to fully validate them in an operational context. Also, as previously mentioned, there is the question of procedure definition and validation.

In view of this, initial deployments of MATIS will probably be used in a success-oriented manner. That is, there will be no automated recovery from anomalies. In this scenario the spacecraft controller would monitor the execution of the schedule on MATIS and resume manual operations if something unexpected happened. This could reduce the need for dedicated spacecraft controllers per mission, as it could permit spacecraft controllers to monitor the execution of schedules for a number of similar missions (families of missions), e.g., Rosetta, MEX, and VEX.

Acknowledgments

The authors wish to acknowledge the valuable input they have received from Y. Doat and E. M. Sørensen in defining the operational concept for automation of ESOC operations outlined in this chapter.

References

[1] *Cross Support Reference Model—Part 1: Space Link Extension Services*, Consultative Committee for Space Data Systems, CCSDS 910.4-B-2, Washington, DC, Oct. 2005.

[2] “Space Engineering: Ground Systems and Operations—Procedure Definition Language,” ECSS-E-70-32, Issue 1.0.

[3] “Space Engineering: Ground Systems and Operations—Monitoring and Control Data Definition,” ECSS-E-70-31.

[4] Haddow, C. R., Müller, C., Scotti, G., and Ulriksen, T., “Network Interface System (NIS); ESA’s Next Generation CCSDS SLE Interface System,” *SpaceOps 2006 Conference*, Rome, June 2006.

Chapter 19

Enhancing Science and Automating Operations Using Onboard Autonomy

Rob Sherwood[*] and Steve Chien[†]

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

I. Introduction

SINCE January 2004, the Autonomous Sciencecraft Experiment (ASE) running on the Earth Observing-1 (EO-1) spacecraft has demonstrated several integrated autonomy technologies to enable autonomous science. Several science algorithms, including onboard event detection, feature detection, and change detection, are being used to analyze science data. These algorithms are used to downlink science data only on change, and detect features of scientific interest such as volcanic eruptions, growth and retreat of ice caps, flooding events, and cloud detection. These onboard science algorithms are inputs to onboard planning software that can modify the spacecraft observation plan to capture high-value science events. This new observation plan is then executed by a robust goal and task-oriented execution system, able to adjust the plan to succeed despite run-time anomalies and uncertainties. Together these technologies enable autonomous goal-directed exploration and data acquisition to maximize science return. This chapter describes the specifics of the ASE and relates it to past and future flights to validate and mature this technology.

The ASE onboard flight software includes several autonomy software components:

1) Onboard science algorithms that analyze the image data to detect trigger conditions such as science events, "interesting" features, changes relative to previous observations, and cloud detection for onboard image masking.

[*]Program Manager, Earth Science Information Systems. Senior Member AIAA.

[†]Principal Investigator, Autonomous Sciencecraft Experiment.

2) Robust execution management software using the Spacecraft Command Language (SCL) [1] package to enable event-driven processing and low-level autonomy.

3) The Continuous Activity Scheduling Planning Execution and Replanning (CASPER) [2] software that replans activities, including downlink, based on science observations in the previous orbit cycles.

The onboard science algorithms analyze the images to extract static features and detect changes relative to previous observations. The software uses EO-1 Hyperion instrument images to automatically identify regions of interest including land, ice, snow, water, and thermally hot areas. Repeat imagery using these algorithms can detect regions of change (such as flooding, ice melt, and lava flows). Using these algorithms onboard enables retargeting and search, e.g., retargeting the instrument on a subsequent orbit cycle to identify and capture the full extent of a flood.

Although the ASE software is running on the EO-1, it will also be used on other interplanetary space missions. On these missions, onboard science analysis will enable capture of short-lived science phenomena. In addition, onboard science analysis will enable data to be captured at the finest time scales without overwhelming onboard memory or downlink capacities by varying the data collection rate on the fly. The software is currently undergoing infusion on the Mars Exploration Rovers mission and Mars Odyssey mission. Examples of future mission applications to use this software include eruption of volcanoes on Io, formation of jets on comets, and phase transitions in ring systems. Generation of derived science products (e.g., boundary descriptions, catalogs) and change-based triggering will also reduce data volumes to a manageable level for extended duration missions that study long-term phenomena such as atmospheric changes at Jupiter and flexing and cracking of the ice crust on Europa.

The onboard planner (CASPER) generates mission operations plans from goals provided by the onboard science analysis module. The model-based planning algorithms enable rapid response to a wide range of operations scenarios based on a model of spacecraft constraints, including faster recovery from spacecraft anomalies. The onboard planner accepts as inputs the science and engineering goals and ensures high-level goal-oriented behavior.

The robust execution system (SCL) accepts the CASPER-derived plan as an input and expands the plan into low-level commands. SCL monitors the execution of the plan and has the flexibility and knowledge to perform event-driven commanding to enable local improvements in execution as well as local responses to anomalies.

II. EO-1 Mission

The EO-1 is the first satellite in NASA's New Millennium Program Earth Observing series [3]. The goal of the EO-1 primary mission was to develop and test a set of advanced technology land imaging instruments. EO-1 was launched on a Delta 7320 from Vandenberg Air Force Base on 21 November 2000. It was inserted into a 705-km circular, sun-synchronous orbit at a 98.7 deg inclination. This orbit allows for 16-day repeat tracks, with between 5 (at the equator to 45 deg) and 16 (at the poles) overflights per 16-day cycle. For each scene, between

13 to as much as 48 Gbits of data from the Advanced Land Imager (ALI), Hyperion, and Atmospheric Corrector (AC) are collected and stored on the onboard solid-state data recorder.

EO-1 is currently in extended mission, having more than achieved its original technology validation goals. As an example, over 27,000 data collection events have been successfully completed, against original success criteria of 1000 data collection events. The ASE described in this chapter uses the Hyperion hyperspectral instrument. The Hyperion is a high-resolution imager capable of resolving 220 spectral bands (from 0.4 to 2.5 μm) with a 30-m spatial resolution. The instrument images a 7.7×42 km land area per image and provides detailed spectral mapping across all 220 channels with high radiometric accuracy.

The EO-1 spacecraft has two Mongoose M5 processors. The first M5 is used for the EO-1 command and data handling functions. The other M5 is part of the wideband advanced recorder processor (WARP), a large mass storage device. Each M5 runs at 12 MHz (for ~8 MIPS) and has 256 MB RAM. Both M5s run the VxWorks operating system. The ASE software operates on the WARP M5. This provides an added level of safety for the spacecraft because the ASE software does not run on the main spacecraft processor.

III. Onboard Science Analysis

The first step in the autonomous science decision cycle is detection of interesting science events. Twelve of the Hyperion spectral bands are used to classify the pixels within each image as land, ice, water, snow, clouds, and fresh lava. Using the pixel classification, a number of science analysis algorithms are being used including:

1) Thermal anomaly detection, which uses infrared spectra peaks to detect lava flows and other volcanic activity. (see Fig. 1).

2) Cloud detection [4], which uses intensities at six different spectra and thresholds to identify likely clouds in scenes (see Fig. 2).

3) Flood scene classification, which uses ratios at several spectra to identify signatures of water inundation as well as vegetation changes caused by flooding (see Fig. 3).

4) Change detection, which uses multiple spectra to identify regions changed from one image to another. This technique is applicable to many science phenomena, including lava flows, flooding, freezing, and thawing, and is used in conjunction with cloud detection, (see Fig. 3).

Figure 1 contains both the visible image and thermal detection at the Kilaeua volcano in Hawaii. The infrared bands are used to detect hot areas that might represent fresh lava flows within the image. In the second third of this image, these hot spots are shown in yellow and orange. The area of hot pixels can be compared with the count of hot pixels from a previous image of the same area to determine if change has occurred. If there has been change, a new image might be triggered to get a more detailed look at the eruption.

Figure 2 shows a Hyperion scene and the results of the cloud detection algorithm [4]. This Massachusetts Institute of Technology (MIT) Lincoln Lab developed algorithm is able to discriminate between cloud pixels and land pixels within an image. Specifically, the grey area in the detection results is clouds while the

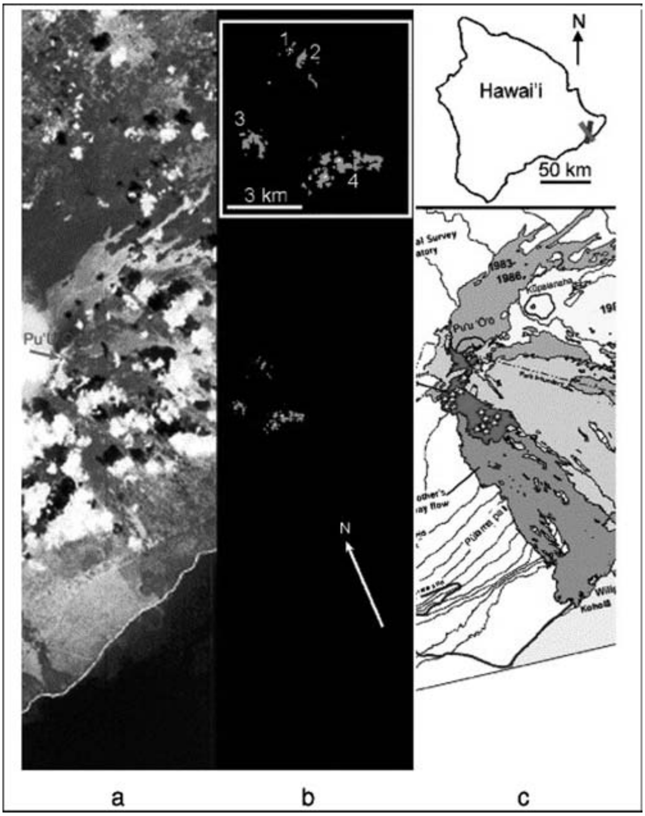


Fig. 1 Kilauea Volcano: a) the visible image of Kilauea, Hawaii, on 24 January 2004; b) thermal classifier output including an inset enlargement of the active area; c) the USGS – Hawaiian Volcano Observatory map showing volcanically active areas in January 2004. Yellow areas delineate the Martin Luther King (MLK) flows in January 2004. (See also the color figure section starting on p. 645.)

blue area is land. The results of this algorithm can be used to discard images that are too cloudy. Images with low cloud cover can be further analyzed for science value.

Figure 3 contains four EO-1 Hyperion images of the Diamantina River in Australia, along with their corresponding classification images to the right of each image. The first image is a baseline image of the river in a dry state. The black area of the corresponding represents all land pixels with no water. The second image two weeks later shows a large flood area with blue representing water pixels. The final two images show the flood receding over time. The results of the algorithm are compared with land and water counts from a previous image to determine if flooding has occurred. If significant flooding has been detected, the image can be downlinked. In addition, a new goal can be sent to the CASPER planning software

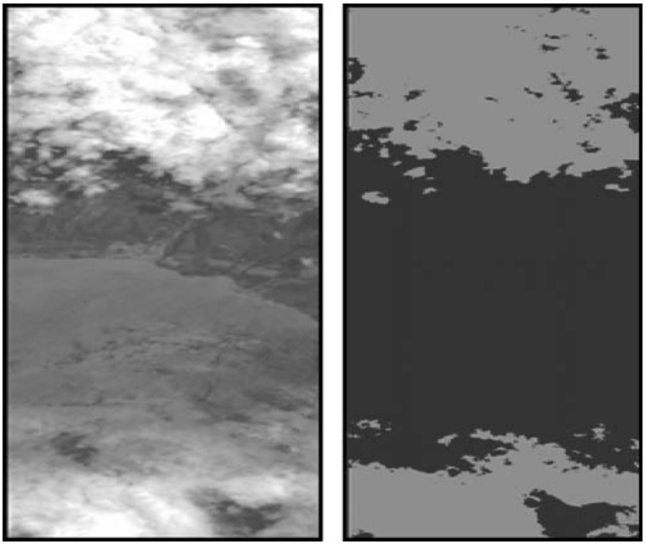


Fig. 2 Cloud detection—visual image at left, grey in right image indicates detected cloud. (See also the color figure section starting on p. 645.)

to image adjacent regions on subsequent orbits to determine the extent of the flooding.

The Jet Propulsion Laboratory (JPL) developed thermal anomaly algorithms use the infrared spectral bands to detect sites of active volcanism. There are two different algorithms, one for daytime images and one for nighttime images. The algorithms compare the number of thermally active pixels within the image with the count from a previous image to determine if new volcanism is present. If no

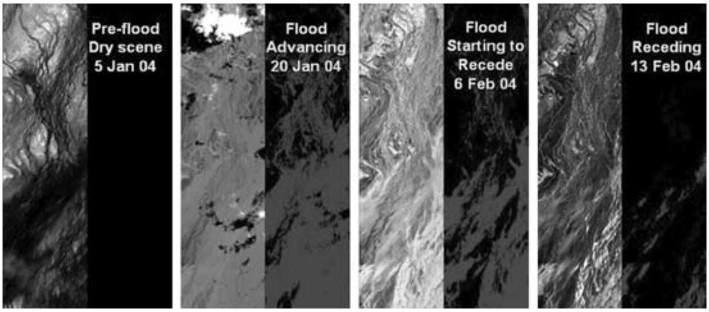


Fig. 3 Flood detection time series imagery of Australia's Diamantina River with visual spectra at left and flood detection map at right. Flooding is caused by monsoonal rain. (See also the color figure section starting on p. 645.)

new volcanism is present, the image can be discarded onboard. Otherwise, the entire image or the interesting section of the image can be downlinked.

The Arizona State University–developed snow-water-ice-land (SWIL) algorithm is used to detect lake freeze/thaw cycles and seasonal sea ice. The SWIL algorithm uses six spectral bands for analysis.

IV. Onboard Mission Planning

For the spacecraft to respond autonomously to the science event, it must be able to independently perform the mission planning function. This requires software that can model all relevant spacecraft and mission constraints. The CASPER [2] software performs this function for ASE. CASPER represents the operations constraints in a general modeling language and reasons about these constraints to generate new operations plans that respect spacecraft and mission constraints and resources. CASPER uses a local search approach [5] to develop operations plans.

Because onboard computing resources are scarce, CASPER must be very efficient in generating plans. While a typical desktop or laptop PC may have 2000–3000 MIPS performance, 5–20 MIPS is more typical onboard a spacecraft. In the case of EO-1, the Mongoose V CPU has approximately 8 MIPS. Of the three software packages, CASPER is by far the most computationally intensive. For that reason, our optimization efforts were focused on CASPER. In light of the performance constraints, we developed an EO-1 CASPER model that did not require a lot of planning iterations. For that reason, the model has only a handful of resources to reason about. This ensures that CASPER is able to build a plan in tens of minutes on the relatively slow CPU.

CASPER is responsible for mission planning in response to both science goals derived onboard as well as anomalies. In this role, CASPER must plan and schedule activities to achieve science and engineering goals while respecting resource and other spacecraft operations constraints. For example, when acquiring an initial image, a volcanic event is detected. This event may warrant a high-priority request for a subsequent image of the target to study the evolving phenomena. In this case, CASPER modifies the operations plan to include the necessary activities to re-image. This may include determining the next overflight opportunity, ensuring that the spacecraft is pointed appropriately, that sufficient power and data storage are available, that appropriate calibration images are acquired, and that the instrument is properly prepared for the data acquisition.

V. Onboard Robust Execution

ASE uses the SCL [1] to provide robust execution. SCL is a software package that integrates procedural programming with a real-time, forward-chaining, rule-based system. A publish/subscribe software bus, which is part of SCL, allows the distribution of notification and request messages to integrate SCL with other onboard software. This design enables both loose or tight coupling between SCL and other flight software as appropriate.

The SCL “smart” executive supports the command and control function. Users can define scripts in an English-like manner. Compiled on the ground, those scripts can be dynamically loaded onboard and executed at an absolute or relative

time. Ground-based absolute time script scheduling is equivalent to the traditional procedural approach to spacecraft operations based on time. In the EO-1 experiment, SCL scripts are planned and scheduled by the CASPER onboard planner. The science analysis algorithms and SCL work in a cooperative manner to generate new goals for CASPER. These goals are sent as messages on the software bus.

Many aspects of autonomy are implemented in SCL. For example, SCL implements many constraint checks that are redundant with those in the EO-1 fault protection software. Before SCL sends each command to the EO-1 command processor, it undergoes a series of constraint checks to ensure that it is a valid command. Any prerequisite states required by the command are checked (such as the communications system being in the correct mode to accept a command). SCL also verifies that there is sufficient power so that the command does not trigger a low bus voltage condition and that there is sufficient energy in the battery. Using SCL to check these constraints and including them in the CASPER model provides an additional level of safety to the autonomy flight software.

VI. Past Operations Flow

The EO-1 spacecraft [3] is operated out of the EO-1 Mission Operations Control Center (MOCC) at the Goddard Space Flight Center (GSFC). The Mission Operations Planning and Scheduling System (MOPSS) was used for long-term planning. The Advanced Spacecraft Integration and System Test (ASIST) tool is used for real-time operations including sending commands and receiving and displaying telemetry. Much of the EO-1 ground and flight systems are similar to the microwave anisotropy probe (MAP) [6] systems. Figure 4 contains the past EO-1 planning and operations flow.

A good approximation of the spacecraft orbit can be predicted about a week in advance. Therefore, the spacecraft schedule of activities was generated on a weekly basis, but because a 1-day orbit prediction is more accurate, the detailed commands were generated and uploaded on a daily basis.

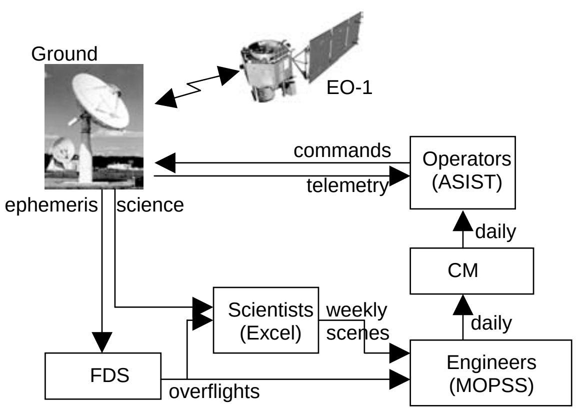


Fig. 4 Pre-autonomy EO-1 operations flow.

A. Past Weekly Operations

In the past, the U.S. Geological Survey (USGS) managed the science requests for EO-1. These included standing and one-time requests from the EO-1 science team, the USGS, and from paying external customers. The first step in operations is to process the long-term plan (LTP) of requests received by the USGS. This plan is a list of targets that would be visible for the upcoming week, including the orbits in which they will be visible.

The ground contact support for EO-1 is managed out of the White Sands Complex (WSC). There are several stations in a ground network (GN) available to EO-1, including the primary sites in Poker Flats, Alaska, and Svalbard, Norway. The next step in operations is to process the GN schedule received by the WSC. This is list of scheduled contacts between EO-1 and the ground stations. Next, the Flight Dynamics Support System (FDSS) at GSFC is used to calculate the spacecraft ephemeris, predicting the spacecraft orbit through the upcoming week. This determines the approximate overflight times for science targets and ground station contacts.

In any given orbit (90 min long), many ground targets are visible, but only about one to two images can be taken due to operations constraints. Therefore, conflicting scenes for a given week must be selected from the list of requests. Scene priorities are based on several factors, including who made the request, if it was paid for, and if it involves a fleeting science event. A USGS representative would manually prioritize and select the scene with the highest priority in a given orbit. Also, the EO-1 science and engineering teams would meet weekly with USGS to verify the selected requests and to make minor modifications to the plan for the following week.

After collecting several scenes, the WARP would reach capacity, and commands must be scheduled to free up space for new requests. Before this can be done, an X-band contact must be scheduled to downlink the science data to Earth. These activities are selected at the same weekly science meeting when images are selected. About one X-band contact every other orbit is selected to keep the WARP from overfilling.

B. Past Daily Operations

After the weekly science meeting, the mission planner would use MOPSS to begin scheduling the one-week set of activities. First, spacecraft maneuver commands are scheduled for each scene. Using the ephemeris, parameter values are calculated for the maneuver commands that will point the instruments toward the target. After each scene is imaged, another maneuver command is added to the schedule to point the spacecraft at nadir. Next, because the maneuvers use reaction wheels, more commands are added to bias and desaturate the wheels. When the wheels change directions, they are less stable and may produce jitter during the observation. Therefore, prior to a group of scenes, the wheels are biased to a non-zero spin rate at the times when data will be collected for the scenes. After a group of scenes, the wheels are desaturated by biasing them to a zero spin rate. This provides the maximum flexibility for spinning the wheels in either direction for subsequent biasing.

While the original manual selection of scenes and contacts are done with the spacecraft requirements in mind, scheduling the details for these activities may still reveal conflicts. MOPSS identifies these conflicts, and the mission planner would need to resolve them manually. When all conflicts are resolved for the next day, the activities are sent from MOPSS to the Command Management System (CMS), where the sequence is generated and prepared for uplink to the spacecraft. The commands for a given day are typically prepared the day before, then uplinked using ASIST on the next available ground contact. This is performed at the latest reasonable time so that the most accurate ephemeris data can be used to generate the command parameters, and because the sequence is difficult to change once it has been loaded onboard.

Replanning for new science requests, while possible, is difficult in this scenario. After executing an original set of requests, the scientists must wait for the image products to be delivered. This includes waiting for the next X-band downlink, and often includes several days of waiting for the data (stored on tape) to be manually shipped from the ground stations. Once the data arrives, the scientists can run any number of manual or automated analyses on the images. The results of the analyses may suggest a change in priorities for the upcoming requests. For example, detecting a fleeting event such as a forest fire may increase the priority of a repeat scene of the same target. This request is then made at the next weekly meeting. However, if the meeting has already occurred, then the change may require manual rescheduling steps, and must be negotiated with the operations team. If the command sequence has already been uploaded, then the change is difficult and typically not worth the risk.

VII. Current Operations Flow

The ASE team developed advanced operations software for the EO-1 mission. Much of this software is used both on ground workstations for mission operations and on the flight processor for autonomous operations. For example, we make use of the Automated Scheduling Planning Environment (ASPEN) [5], the ground version of the onboard CASPER planner. In this section, we discuss the impact of this software on the weekly and daily operations of EO-1. Figure 5 contains a flow diagram of the current EO-1 operations process including the ASE software. Table 1 contains a comparison of the previous operations steps with the modified steps that include ASE.

A. Weekly Operations

For the weekly science planning, ASPEN is used to lighten the workload of the scientists and engineers. Rather than *selecting* science targets, which requires knowledge of the spacecraft operations constraints, the scientists need only to *prioritize* the LTP for the upcoming week. ASPEN is then used to select the highest priority scenes while respecting spacecraft and operations constraints.

ASPEN is also used to schedule downlinks for the observations. The GN schedule identifies the set of X-band downlink opportunities and ASPEN uses its model of the WARP to predict when the memory will reach capacity. Using this model, it automatically adds X-band downlinks and file delete activities to free up space

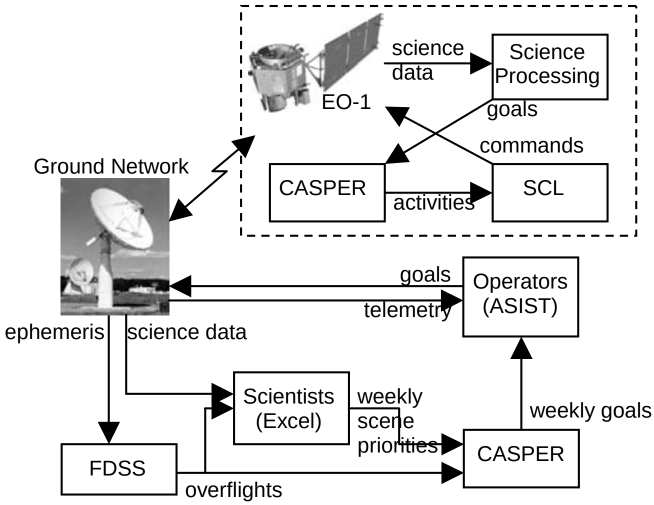


Fig. 5 EO-1 operations flow with ASE.

Table 1 Comparison of operations process steps before and after ASE

Step	Current operations	Modified operations
1	Process LTP requests	(same)
2	Process JPL requests and convert to EO-1 LTP	(same)
3	Process GN schedule	(same)
4	Process ephemeris and overflights	(same)
5	Manually prioritize science targets	(same)
6	Manually select science targets	ASPEN selects science targets
7	Manually schedule downlinks	ASPEN schedules downlinks
8	Manually schedule maneuvers	ASPEN schedules maneuvers
9	Manually schedule momentum wheel commands	ASPEN schedules momentum wheel commands
10	Generate sequence and uplink	Uplink high-level goals
11	Load time-tagged sequence into onboard queue	CASPER loads goals and generates plan
12	Execute sequence	SCL executes and monitors sequence
13	Manually reprioritize science targets	Science algorithms reprioritize science targets
14	Manually select replacement targets	Science algorithms select replacement targets
15	Manually reschedule	CASPER reschedules
16	Generate sequence and uplink	No uplink required

on the recorder. As with all activities, these are scheduled where allowed by the overflights and spacecraft constraints.

Finally, ASPEN interfaces with the FDSS and generates the required maneuver and wheel bias commands. The FDSS software uses the spacecraft ephemeris to provide the required parameters for the commands. The ephemeris file is generated using estimates of the spacecraft position and velocity vectors. Using GPS data, these vectors are calculated onboard for the ACS, but the velocity vectors are not accurate enough to make long-term predictions. Therefore, weekly predictions are made on the ground using tracking data from the GN.

B. Daily Operations

With the planner operating onboard the spacecraft, we do not need to uplink the detailed command sequence, but only the high-level requests for scenes, downlinks, maneuvers, and wheel biasing. When CASPER receives these goals, it will expand them into more detailed activities and schedule all activities at non-conflicting start times. Pending activities are continuously sent to SCL, where the appropriate commands are executed and monitored.

This also means that we can uplink the entire week of goals rather than one day at a time. However, as the estimates for orbital parameters change, we need to send commands to the planner to change the relevant parts of the plan. This includes changes to scene start times and to parameters for maneuvers and wheel bias activities. When the planner receives these commands, it makes these and other changes necessary to maintain consistency in the plan. Using the ephemeris and the FDSS on the ground, these commands must be uplinked, presumably on a daily basis. However, the onboard ephemeris is accurate enough for 1-day predictions and could be used to generate these daily plan updates. This would require additional work to port the FDSS (currently implemented in MATLAB®) to the flight processor and operating system. Another alternative would be to skip the daily updates and use the less accurate (generated weekly) parameters for pointing, biasing, and image timing. This would result in slightly degraded science data, but possibly still within acceptable limits. Ultimately, this work would close the loop and allow us to fly autonomously for a full week. The final decision on the actual implementation will depend on available project resources.

Replanning scenarios become much easier with the addition of the ASE flight software. First, the science products are immediately available onboard after executing a scene request. The onboard science algorithms can start analyzing the data much earlier than if the analysis were done on the ground. The results of the analysis can then trigger new requests, which are immediately sent to CASPER onboard. After receiving the new requests, CASPER will change the plan to accommodate the requests while maintaining consistency with spacecraft constraints. Onboard analysis and replanning takes only minutes compared to ground-based operations, which may take days.

To replan science activities onboard, we also need to replan the associated maneuver and wheel bias activities that were originally planned on the ground. The parameters for these activities, calculated by the FDSS, depend on the prior spacecraft orientation and wheel speed. However, by making a few simple assumptions, these activities can be scheduled onboard without requiring parameter

recalculations. Specifically, if we assume that we always slew to nadir after a scene, then all maneuvers will begin at nadir and the parameters will remain constant regardless of the order of the scenes. If we also assume that the wheels are biased prior to each scene and desaturated after each scene, again, the parameters remain constant. Therefore, CASPER can change the plan in-flight using values precalculated by the FDSS. The disadvantage is that the plan will contain unnecessary activities and may not be optimal. We have not actually analyzed the quantity of extra activities, but the operations team believes it has no significant impact on the satellite operations lifetime, and would introduce a lot of added complexity into the operations flow to eliminate. It is also worth noting that we have a much reduced operations budget.

VIII. New Ground Software

Additional ground support software has been put in place to integrate the ASE architecture into EO-1 operations procedures. This software package interfaces with the science and operations teams to coordinate the selection of observations, pre-flight testing, and post-flight data management. A web-based interface manages each of the following steps:

1) Generating a list of potential observations for the upcoming week.
2) Providing an interface for the ASE science team to select observations and science analysis parameters.
3) Converting the selected observations to CASPER science goals.
4) Validating the autonomous execution of these observations on the ground test beds.
5) Sending the validated goals to the EO-1 operations team for uplink to ASE on EO-1.
6) Testing.
7) Processing and validating the telemetry and science data returned from the autonomous execution of the science goals onboard EO-1.
8) Logging and cataloging the products from each operations step.
9) Sending e-mail notifications to the relevant personnel for each step.

Additionally, a web-based interface was constructed to enable EO-1 operations personnel to request engineering activities for the generated schedules. These include instrument calibrations, instrument outgassing activities, and maneuvers. With the exception of maneuvers, each of these activities is scheduled and expanded out to appropriate commands by the automation software. For maneuvers, ASPEN blocks out an appropriate time window and the flight dynamics team generates the exact command sequence required for orbit maintenance and it is inserted into the schedule.

IX. EO-1 Operational Cost Savings as a Result of Autonomous Science

The ASE software has been flying onboard EO-1 from 2003 to 2006. Both flight and ground elements of the software have enabled large portions of the observation planning, sequence generation, and command load uplink processes to be automated for EO-1, resulting in considerable cost savings while maintaining

Table 2 Annual cost and savings for EO-1 operations

Operations activity	Cost before ASE, $K	Cost using ASE, $K	Savings, $K
Mission planning and sequence generation (reduction in personnel)	800	200	600
Observation planning	1140	720	420
Total	1940	920	$1020K

or improving mission science return. To compute the cost savings from ASE automation, the following steps were performed:

1) The cost for operating the EO-1 mission for basic science capability but without the ASE software were estimated. This basic science capability included the averaging of approximately 100 observations per week, the ability to routinely retask to priority targets with 24 h (or less) notice, and the retention of the basic spacecraft observation capability (e.g., minimal risk to the spacecraft, reasonable use of consumables).

2) The cost for operating the EO-1 mission using ASE to acquire 100 observations per week was calculated. Further ground automation in the mission planning process was enabled by using ASE, which contributed to a lower operations cost.

The cost savings are the difference between steps 1 and 2. In estimating both steps 1 and 2, the EO-1 Flight Operations Team (FOT) developed estimates by a bottom-up calculation by the doing functions (e.g., flight dynamics, operators, sequence generation). Whenever possible the computation of step 1 was derived from actual operations expenditures in the prior operations of EO-1. The computation of step 2 was based on the experience of operating ASE through September 2005. Because these figures are derived from actual expenditures, they are considered very accurate.

Mission planning and sequence generation savings are due to automation of portions of the baseline schedule generation, ground tracking station allocation, sequence generation, command load generation, and uplinking of goal files that replace command loads. Observation planning savings are due to automating portions of the observation selection process. Other indirect savings have been enabled. Because goal file uploads occur automatically, command files are no longer needed to be uploaded on weekends—enabling reduction of weekend operator staffing (2 days per week). These cost savings have been validated with the continuing operations of EO-1 during FY 2006 (e.g., from 1 October 2005 through 1 February 2006). Operations of the EO-1 satellite during this period have been performed within the described budget, and science return has met or exceeded the guideline requirements.

For the specific case of using the ASE software on EO-1, science return per data downlink was increased by over 100× by rapid response and returning the most important science data (see Table 3).

For specific cost savings (or value added) from increased science return from ASE, we submit the following analysis. To compute an economic value to the

Table 3 Downlink data savings by science process

Process	Total process data acquired	Data returned by ASE	Downlink savings	Savings factor (goal was ×10)
Volcanism	33750 MB	294 MB	33456 MB	115
Cyrosphere (ice)	38100 MB	304 MB	37796 MB	125
Flooding	25500 MB	239 MB	25261 MB	106
Total	97350 MB	837 MB	96513 MB	116

baseline EO-1 science return, we use a conservative estimate based on the minimum ($1000/image) cost for scenes, along with the typical number of paid images per day (8), and a conservative estimate of science operations days per month (some are lost for engineering operations):

$1000/image × 8 images/day × 25 days/month × 12 months/year = $2.4M/year

We take this as a conservative estimate of the value of the science return from conventional EO-1 operations. Assuming a conservative science increase of 10× (compared to the documented increase of over 100×), ASE has increased the science return of EO-1 as follows:

science return with ASE – science return without ASE
= 10× $2.4M/yr – 1× $2.4M/yr = $21.6M/yr

X. Technology Validation and Flight Status

ASE started as a technology experiment. Prior to uploading the software to the EO-1 spacecraft, the software was run through an extensive testing program on several ground-based test beds. These were low-fidelity test beds that used software simulators for the spacecraft instruments. As a result, we had to run several tests onboard EO-1 to demonstrate the capabilities of ASE prior to running the technology validation experiments. We slowly built up the autonomy capability by testing each component separately before running an integrated systems test.

The technology was declared fully validated in May 2004 after all 20 onboard autonomy experiments were fully tested. The overall system performed as expected and was considered a success. The validation consisted of the following onboard autonomy experiments run 5 times each: 1) image planning and acquisition, 2) downlink, 3) data editing, and image acquisition followed by image retargeting.

Since the completion of the technology validation, over 4000 more autonomous data acquisitions have been completed. In addition, we have run over 400 closed-loop executions where ASE autonomously analyzes science data onboard and triggers subsequent observations. The software has been running full-time onboard the EO-1 satellite for the past several months. ASE is now the primary mission planning and control system.

There were two important risks to our technology validation approach—one technical and one political. The technical risk was related to spacecraft safety. If the EO-1 satellite was lost due to the ASE software, that would have been a

huge setback for onboard spacecraft autonomy. This risk was mitigated using three different methods. First, we had an extensive testing program to ensure that the software would operate as expected. Second, we had triple redundancy built into the three-layered architecture of this autonomy software. Lastly, we ran the software on the solid-state recorder CPU (WARP) rather than the main spacecraft CPU.

The second risk was political. We needed to ensure that the technology validation of our software was convincing enough that scientists would use it on future missions. We had a multifaceted approach to achieve this goal. First and foremost, we involved (and funded) several scientists in the development of the experiment, software, and operations of the ASE software. The idea is that if the scientists are involved from the start, they will help us develop a useful system and they will promote it to their peers. Another method we employed to ensure future use was to go way beyond the minimal set of validation experiments to show that this software is durable, maintainable, and can achieve increased science. We also started technology infusion early. This effort has so far paid off with infusion under way into the Mars Odyssey and Mars Exploration Rover missions.

XI. EO-1 Sensorweb

The use of automated planning onboard EO-1 has enabled a new system-of-systems capability. We have networked the EO-1 satellite with other satellites and ground sensors. This network is linked by software and the Internet to an autonomous satellite observation response capability. This system is designed with a flexible, modular architecture to facilitate expansion in sensors, customization of trigger conditions, and customization of responses.

The EO-1 sensorweb has been used to implement a global surveillance program of science phenomena, including volcanoes, flooding, cryosphere events, and atmospheric phenomena. Using this architecture, we have performed over 700 sensorweb initiated satellite observations using EO-1. The automated retasking element of the sensorweb consists of several components working together as follows:

1) Science agents for each of the science disciplines automatically acquire and process satellite and ground network data to track science phenomena of interest. These science agents publish their data automatically to the internet each in their own format. In some cases this is via the http or ftp protocol, in some cases via e-mail subscription and alert protocols.

2) Science agents either poll these sites (http or ftp) to pull science data or simply receive e-mails to receive notifications of ongoing science events. These science agents produce “science event notifications” in a standard XML format, which are then logged into a “science event” database.

3) The science event manager processes these science event notifications and matches them up with particular science campaigns as defined by the scientists. When a match occurs, an observation request is generated.

4) These observation requests are processed by the ASPEN automated mission planning system. ASPEN integrates these requests and schedules EO-1 observations according to priorities and mission constraints.

5) For observations that are feasible, an observation request is uplinked to the spacecraft.

6) Onboard EO-1 the ASE software will accommodate the observation request if feasible. In some cases onboard software may have additional knowledge of spacecraft resources or may have triggered additional observations, and so some uplinked requests may not be feasible.

7) Later, the science data are downlinked, processed, and delivered to the requesting scientist.

The science agents encapsulate sensor and science tracking specific information by producing a generic XML alert for each "science event" tracked. The flexibility enabled by these modules allows us to easily integrate with a large number of existing science tracking systems despite the fact that each science tracking system has its own unique data and reporting format. The data formats range from near raw instrument data, to alerts in text format, to periodic updates to a wide range of text formats. The posting methods include http, https, ftp, and e-mail.

The science event manager enables scientists to specify mappings from science events to observation requests. It enables them to track frequency and count of events and do logical processing. It also enables them to track based on target names or locations, and other event-specific parameters (for example, some tracking systems produce a confidence measure). As an example, a volcanologist might specify for the Kilauea site that several tracking systems would need to report activity with high confidence before an observation is requested. This is because Kilauea is quite often active. On the other hand, even a single low-confidence activity notification might trigger observation of Piton de la Fournaise or other less active sites.

A. Wildfire Sensorweb

We have demonstrated the sensorweb concept using the Moderate Resolution Imaging Spectroradiometer (MODIS) active fire mapping system. Both the Terra and Aqua spacecraft carry the MODIS instrument, providing morning, afternoon, and two night overflights of each location on the globe per day (coverage near the poles is even more frequent). The active fire mapping system [7] uses data from the GSFC Distributed Active Archive Center (DAAC), specifically the data with the predicted orbital ephemeris, which is approximately 3–6 h from acquisition. Figure 6 shows the active fire map from October 2003 fires in Southern California. The active fire alerts from the MODIS data are used to trigger higher resolution EO-1 images.

B. Flood Sensorweb

The flood sensorweb uses the Dartmouth Flood Observatory (DFO) Global Active Flood Archive to identify floods in remote locations automatically based on satellite data. The DFO flood archive publishes web-based flood alerts based on MODIS, QuikSCAT, and AMSR-E [8] satellite data. The flood sensorweb utilizes the DFO QuikSCAT atlas because it is not affected by cloud cover over flooded areas [9].

In the flood sensorweb, we target potential areas of flooding at gauging reaches. Gauging reaches are river locations whose topography is well understood. Flood discharge measurements at gauging reaches can be used to measure the amount of

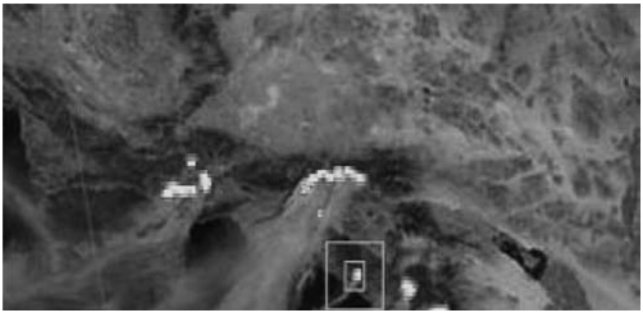


Fig. 6 Active fire alerts for the October 2003 Southern California fires. Red indicates active fires. The light blue box illustrates the background region used in the relative threshold detection. (See also the color figure section starting on p. 645.)

water passing through a flooded region and can be compared with remotely sensed data. The end effect of the flood sensorweb is to increase the amount of high resolution (EO-1) remote sensing data available on flooding events in prime locations of interest (e.g., gauging reaches) and times of interest (e.g., when active flooding occurs). Imagery from an August 2003 flood sensorweb demonstration capturing flooding in the Brahmaputra River, India, is shown in Fig. 7.

C. Volcano Sensorweb

In the volcano sensorweb, MODIS, Geostationary Operational Environmental Satellites (GOES7), and the Advanced Very High Resolution Radiometer (AVHRR) sensor platforms are utilized to detect volcanic activity [11]. These alerts are then used to trigger EO-1 observations. The EO-1 Hyperion instrument

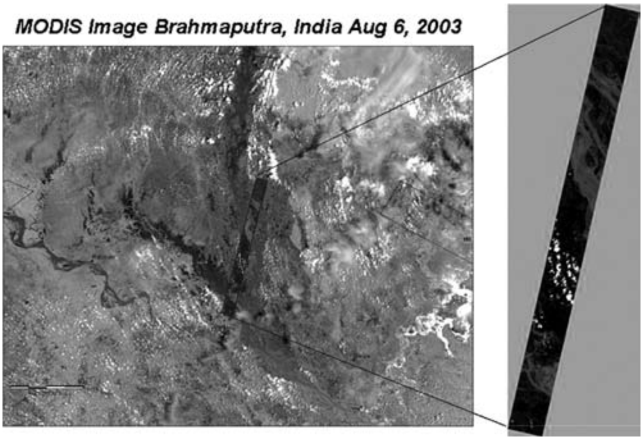


Fig. 7 Examples of 250-m low-resolution MODIS imagery (left) and 30-m EO-1 imagery (right) from the flood sensorweb capturing Brahmaputra River flooding in India, August 2003. (See also the color figure section starting on p. 645.)

is ideal for study of volcanic processes because of its great sensitivity range in the infrared spectrum.

The GOES [11] and AVHRR alert systems provide excellent temporal resolution and rapid triggering based on thermal alerts. The GOES-based system looks for locations that are hot, are high contrast from the surrounding area, and are not visibly bright. Additionally, hits are screened for motion (to eliminate cloud reflections) and persistence (to remove instrument noise). The GOES alert can provide a Web or e-mail alert within 1 h of data acquisition.

We have also linked into in-situ sensors to monitor volcanoes. The Hawaiian Volcano Observatory (HVO) has deployed numerous instruments on the Kilauea region in Hawaii. These instruments include tiltmeters, gas sensors, and seismic instrumentation. These sensors can provide indications that collectively point to a high-probability, near-term eruption, thereby triggering a request for high-resolution, EO-1 imagery. The University of Hawaii has also deployed infrared cameras [12] to a number of volcanic sites worldwide (e.g., Kilauea, Hawaii; Erte Ale, Ethiopia; Sourfiere Hills, Montserrat; Colima and Popocatepetl, Mexico). These infrared cameras can provide a ground-based detection of lava flows based on thermal signatures, thereby alerting the sensorweb.

D. Cryosphere Sensorweb

Many freeze/thaw applications are also of interest. This includes the phenomena of glacial ice breakup, sea ice breakup, melting, and freezing, lake ice freezing and thawing, and snowfall and snowmelt. Using QuikSCAT data, we are tracking snow and ice formation and melting and automatically triggering higher resolution imaging such as with EO-1.

In collaboration with the Center for Limnology of the University of Wisconsin at Madison, we have linked into data streams from the Trout Lake station to use temperature data to trigger imaging of the sites to capture transient freezing and thawing processes.

XII. Technology Infusion

The science component of the ASE software will soon be used onboard the Mars Exploration Rovers (MER) mission to enable onboard detection and summarization of atmospheric events (dust devils and clouds). Recent explorations on the Martian surface have revealed an environment far more dynamic than previously believed. In particular, the atmosphere of Mars is very dynamic. Dust devils and clouds are dynamic atmospheric features that have been observed by the MER. These high science value events have been the subject of considerable study. Both dust devil and cloud detection campaigns have been conducted, but in general these are rare events. For example, only around 10–25% of the cloud campaign images collected have clouds in them. Prior campaigns have involved collecting images at fixed times for return to Earth. This is an inefficient use of downlink bandwidth as the majority of images do not contain dust devils or clouds.

To improve the effectiveness of atmospheric imaging campaigns, we have developed a different approach. In this approach onboard processing is used to screen images for the science features of interest (i.e., clouds and dust devils).

Using this approach, many images can be collected onboard resulting in a much greater time range for capturing the rare phenomena. Even when the images cannot be downlinked (such as when too many events are detected), compact summary statistics on the number and type of events can still be downlinked to provide valuable information. The code has been integrated with the MER flight software, and was uploaded in July 2006.

The science component of the ASE software is also under development for the Mars Odyssey Mission to enhance science return from the Thermal Emission Imaging System (THEMIS) instrument with planned operational capability in the 2nd extended mission (beginning in Fall 2006). In this application, the ASE software will 1) track the seasonal variation in the CO_2 ice caps, 2) detect thermal anomalies, 3) track dust storms, and 4) track Martian clouds.

The MER THEMIS instrument is powered on almost 100% of the time although only 5% of the data are collected due to bandwidth limitations. Using the ASE science algorithms, the THEMIS images can be analyzed onboard for the existence of thermal anomalies, dust storms, and clouds. Only the images that contain these events will be returned to Earth. This will allow the Odyssey Science Team to make use of the other 95% of the data that are currently lost. Detecting thermal anomalies would be a very low probability event but of very high science value. Also, the boundaries of the CO_2 ice caps can be detected, and only the image of the boundary will be returned.

In addition, we are researching autonomous science and sensorweb applications for magnetosphere events for space weather, change detection on Io and Europa, and storm tracking on Jupiter.

XIII. Conclusion

In 1999, the Remote Agent experiment (RAX) [13] executed for a few days onboard the NASA Deep Space One mission. RAX is an example of a classic three-tiered architecture [14], as is ASE. RAX demonstrated a batch onboard planning capability (as opposed to CASPER's continuous planning), and RAX did not demonstrate onboard science.

We learned some important lessons related to the processing capability for using autonomous science onboard. The EO-1 spacecraft contains two very limited capability Mongoose V computers. There were two issues related to these computers. First, it was very advantageous from a safety standpoint that the autonomy software was not running on the main flight control computer. Any problems resulting from the failure of the ASE software will generally result in lost data collects—a much more tolerable failure that bringing down the flight control computer. The second issue was the limited performance (8 MIPS) of the EO-1 CPUs. The low performance necessitated creating simplified planning models so that the planning could be performed in a reasonable time. In addition, we were not able to use previously developed computationally intensive generalized science algorithms.

The performance issue also extends to the infusion targets. The Mars Odyssey and MER CPUs do not have enough performance and memory to run the planning component of ASE. Other future missions will have data-intensive instruments such as radar. In these cases, it may make sense to use a dedicated CPU/FPGA combination to process the data and run the autonomous science software. This

new architecture also will allow image registration for change detection as well as more complicated machine learning science algorithms.

ASE on EO-1 demonstrates an integrated autonomous mission using onboard science analysis, replanning, and robust execution. The ASE performs intelligent science data selection that leads to a reduction in data downlink. In addition, the ASE increases science return through autonomous retargeting. Demonstration of these capabilities onboard EO-1 will enable radically different missions with significant onboard decision making leading to novel science opportunities. The paradigm shift toward highly autonomous spacecraft will enable future NASA missions to achieve significantly greater science returns with reduced risk and reduced operations cost. We also have described ongoing work to link together automated science event tracking system with an autonomous response capability based on automated planning technology. Demonstration of these sensorweb capabilities will enable fast responding science campaigns and increase the science return of spaceborne assets.

Acknowledgments

Portions of this work were performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. We would like to acknowledge the important contributions of Daniel Tran, Ashley Davies, Rebecca Castaño, Gregg Rabideau, Nghia Tang, and Michael Burl of JPL; Dan Mandl, Stuart Frye, Seth Shulman, and Stephen Ungar of GSFC; Jerry Hengemihle and Bruce Trout of Microtel LLC; Jeff D'Agostino of the Hammers Corp.; Robert Bote of Honeywell Corp.; Jim Van Gaasbeck and Darrell Boyer of ICS; Michael Griffin and Hsiao-hua Burke of MIT Lincoln labs; Ronald Greeley and Thomas Doggett of Arizona State University; and Victor Baker and James Dohm of the University of Arizona.

References

[1] Interface and Control Systems, SCL Technical Overview, http://www.interfacecontrol.com/papers/techoverview/techoverview.html [retrieved 10 May 2007].

[2] Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G., "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

[3] NASA Goddard Space Flight Center, EO-1 Mission page, http://EO-1.gsfc.nasa.gov [retrieved 10 May 2007].

[4] Griffin, M., Burke, H., Mandl, D., and Miller, J. "Cloud Cover Detection Algorithm for the EO-1 Hyperion Imagery," *Proceedings of the 17th SPIE AeroSense 2003,* Orlando, FL, April 21–25, 2003.

[5] Rabideau, G., Knight, R., Chien, S., Fukunaga, A., and Govindjee, A. "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," *International Symposium on Artificial Intelligence Robotics and Automation in Space*, Noordwijk, The Netherlands, June 1999.

[6] Breed, J., Coyle, S., Blahut, K., Dent, C., Shendock, R., and Rowe, R. "The MAP Autonomous Mission Control System," *Proceedings of the Sixth International Conference on Space Operations*, Toulouse, France, June 2000.

[7] Justice, C. O., Giglio, L., Korontzi, S., Owens, J., Morisette, J. T., Roy, D., Descloitres, J., Alleaume, S., Petitcolin, F., and Kaufman, Y. "The MODIS Fire Products," *Remote Sensing of Environment*, Vol. 83, No. 1–2, 2002, pp. 244–262.

[8] Nghiem, S. V., Liu, W. T., and Xie, X., "Polarization Reversal over Flooded Regions and Applications to Flood Mapping with Spaceborne Scatterometers," *International Geoscience and Remote Sensing Symposium,* Hamburg, Germany, June 28–July 2, 1999.

[9] Brakenridge, G., Nghiem, S., Anderson, E., and Chien, S. "Space-Based Measurement of River Runoff," *EOS*, Vol. 86, No. 19, May 2005.

[10] Wright, R., Flynn, L. P., Garbeil, H., Harris, A. J. L., and Pilger, E. "MODVOLC: Near-Real-Time Thermal Monitoring of Global Volcanism," *Journal of Volcanology and Geothermal Research,* Vol. 135, 2004, pp. 29–49.

[11] Harris, A., Pilger, E., and Flynn, L. P. "Web-based Hot Spot Monitoring using GOES: What it is and How it Works," *Advances in Environmental Monitoring and Modelling*, Vol. 1, No. 3, 2002, pp. 5–36.

[12] Harris, A., Johnson, J., Horton, K., Garbeil, G., Pirie, D., Ramm, H., Donegan, S., Pilger, E., Flynn, L., Mouginis-Mark, P., Ripepe, M., Marchetti, E., and Rothery, D., "Ground-based Infrared Monitoring Provides New Tool for Remote Tracking of Volcanic Activity," *EOS*, Vol. 84, No. 40, 2003, p. 409.

[13] NASA Ames, Remote Agent Experiment Home Page, http://ic.arc.nasa.gov/projects/remote-agent/ [retrieved 10 May 2007].

[14] Gat, E., "Three-Layer Architectures" *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems,* edited by D. Kortenkamp, R. P. Bonasso, and R. Murphy, MIT Press, Cambridge, MA, 1998, pp. 195–210.

This page intentionally left blank

Chapter 20

Dynamic Allocation of Resources to Improve Scientific Return with Onboard Automated Replanning

Fabrício de Novaes Kucinskis* and Maurício Gonçalves Vieira Ferreira†

Instituto Nacional de Pesquisas Espaciais, São José dos Campos, Brazil

I. Introduction

THE satellites of the Brazilian Program for Scientific Satellites and Experiments, run by the Instituto Nacional de Pesquisas Espaciais (INPE, the Brazilian National Institute for Space Research), carry as payload a set of scientific and technological experiments, which have pre-defined quotas of resources (power and memory, for example) allocated by the onboard computer as the mission engineers had defined it, still in the phase of systems specification. As a result, the experiments collect data through operation plans that generally follow a repetitive pattern.

There are, however, short-duration scientific phenomena of which occurrences, although predictable, are random—an ionospheric disturbance, for instance, can take place at any time and last from minutes to hours. To better analyze these phenomena, it may be important to increase the acquisition rate and/or the precision of the data collected by an experiment. This increases the consumption of power and memory beyond that originally defined.

Because of the short duration and the difficulty in specifying exactly when a phenomenon of this kind will occur, it is not enough to leave the ground operations team in charge of the satellite reconfiguration. The necessary time for the phenomenon to be reported and for the ground team to create and send a new operation plan to the satellite is, in general, much longer than the duration of the phenomenon. In this case, the scientific opportunity to adequately analyze it will have been lost.

---

*Computer Engineer, Technologist of the Aerospace Electronics Division's Onboard Data Handling Group.

†Systems Analyst, Computer Science Researcher of the Satellite Control Center.

There is then the need for allowing the experiment, when detecting the occurrence of a short-duration phenomenon, to request from the onboard computer the temporary reallocation of resources to be able to carry out a more detailed analysis, in such a way as to affect as little as possible the operation of the other experiments and the satellite itself. However, the use of classical programming techniques is not adequate to deal with the great number of states in which the several satellite subsystems and experiments can exist.

In this context, the artificial intelligence (AI) planning and scheduling techniques are presented as a potential solution to be explored, and one of the most promising technologies able to increase satellite autonomy. However, despite its power, the onboard implementation of planning encounters two great obstacles: the limited processing power of satellite computers and the resistance of mission managers and engineers concerning the increase of autonomy.

The challenge lies in allowing the increase of autonomy, without diminishing confidence in the satellite behavior, and in consuming available computer resources as little as possible during the plan process. To achieve both goals, it is necessary to treat planning as an integral part of onboard software, just like telemetry processing, housekeeping, and diagnostic tasks. We are working on a solution that allows the integration of onboard planning with the regular satellite onboard software through a number of issues:

1) The use of the same programming language to develop the onboard software and to describe the satellite model used by the planner.

2) A form of modeling closer to the real satellite operation than other planning approaches.

3) Modules for problem composition and for the planning process that are well integrated with the rest of the satellite software, but that can be removed without harm to the satellite functioning.

4) A safe and gradual form of implementation, based on the gain of confidence of engineers, operators and mission managers.

The next topic is an overview of AI planning and scheduling systems in space missions. Then this chapter describes the concepts, components, and dynamics of our solution in the context of our case of study, the Equatorial Atmosphere Research Satellite (EQUARS), as well as the safe and gradual approach foreseen to validate this technology in INPE's future satellites.

II. AI Planning and Scheduling in Space Missions

According to Fukunaga et al. [1], planning is the selection and sequencing of activities in such a way that they achieve one or more goals and satisfy a set of domain constraints. Zweben et al. [2] define scheduling as the process of assigning times and resources to tasks of a plan, also satisfying a set of domain constraints. A less formal definition, but certainly more clear, was given by Myers and Smith [3]: "by planning we refer generally to the process of deciding *what* to do; [...] we use the term scheduling generally to designate the process of deciding *when* and *how*."

Planning techniques date back from the early 1970s with the Stanford Research Institute Problem Solver (STRIPS) planner [4], but it was the addition of scheduling concepts, such as resource consumption, time assignments, and constraint

satisfaction problems (CSP), that gave planners the ability to deal with real-world problems. The union of them is known as artificial intelligence planning and scheduling, but for the sake of clarity, we will use AI planning or just planning in this chapter.

The search for consistent plans that is the purpose of AI planning fits perfectly the operations routine of space missions. NASA is the agency that has the greatest experience in the development and use of AI planning systems for mission control. The first of them was Science Planning Intelligent Knowledge Environment/ Science Planning and Scheduling System (SPIKE/SPSS) [5], used since the early 1990s to create observation plans for the Hubble Space Telescope. SPIKE was followed by many planners, such as Gerry/Ground Processing Scheduling System (GERRY/GPSS) [2], used for the ground maintenance plan of the space shuttles; HSTS, also developed for Hubble; Data Chaser Automated Planning System (DCAPS) [6], an experiment that generated operation plans for the shuttle's data chaser payload; and finally Automated Scheduling and Planning Environment (ASPEN) [7], an evolution of DCAPS used in many current missions.

ESA also has some history with AI planning. Their first experiences, also in the beginning of the 1990s, were the plan-ERS1 [8], used to generate operation plans for the Earth Resources Satellite One (ERS-1), and Optimum-AIV [9], used to help the assembly, integration, and verifying process of equipments for the Ariane IV rockets. More recently, ESA supported work that created the Mars Express Architecture (MEXAR) I and II [10] planners for the Mars Express mission.

In all of these planners, the plans are generated and validated on ground and just then sent to the spacecraft, as a series of time-tagged telecommands. There are cases, however, in which it is desirable to change plans aboard the spacecraft to increase its autonomy and allow it to have a quick response to external events. This is called onboard replanning, and there are only two reported cases up to now, both from NASA.

In May 1999, the Deep Space One (DS-1) probe was operated for some days through detailed operation plans generated aboard the spacecraft from high-level commands sent by the ground operations team [11]. The experiment, called Remote Agent Experiment (RAX), was considered a great success.

More recently, from October 2003, the remote sensing satellite Earth Observing One (EO-1) started to execute the Autonomous Sciencecraft Experiment (ASE), of which the Continuous Activity Scheduling, Planning, Execution and Replanning (CASPER) planner, an onboard version of ASPEN, is part [12]. CASPER is responsible for replanning the satellite operations to respond to the detection of events of scientific interest, such as floods and volcanic eruptions, which increased the scientific return. The ASE implementation was gradual, and in April 2005, the ground operations team was already using it in normal tasks.

These missions place NASA as the only agency to use planning aboard its spacecrafts. ESA has also been investing in increase of the autonomy with projects such as Project for On-Board Autonomy (PROBA) spacecraft [13], but without the use of onboard planning.

INPE has recently started its line of studies in AI planning, which focus on both the plans generated on ground [14, 15] and onboard planning [16]. The onboard planning research is being done inside a project for the development of a new

onboard computer, called COMAV (from the Portuguese acronym for "Advanced Computer").

III. COMAV and the RASSO Onboard Service

In addition to its satellites programs, INPE maintains many technological development projects managed by different teams spread over the institute. Among these projects is COMAV, which is being developed by the Onboard Data Handling Group (SUBORD) of the Aerospace Electronics Division.

COMAV is a research project for the development of a new onboard computer (OBC) for INPE's future space applications. The first studies of this project are from 2000, and its main purpose is to study new space technologies, components, techniques, and standards while developing an OBC and thus help in the adoption of them in the institute engineering environment. Among COMAV attributions are control and communication with scientific and technological experiments.

The COMAV software is being developed in C language over the RTEMS operating system, distributed under the GNU General Public License terms. The onboard replanning is being implemented as part of a service that provides more resources than originally programmed for experiments that detect the occurrence of scientific short-duration phenomena. This service was given the name Resources Allocation Service for Scientific Opportunities, or just RASSO.

Figure 1 shows the RASSO architecture. The arrows indicate the data flow between the modules during the planning process. A brief description of the service functioning follows.

As shown in Fig. 1, an experiment that detects the occurrence of a short-duration phenomenon sends a request for more resources to RASSO, to make a better

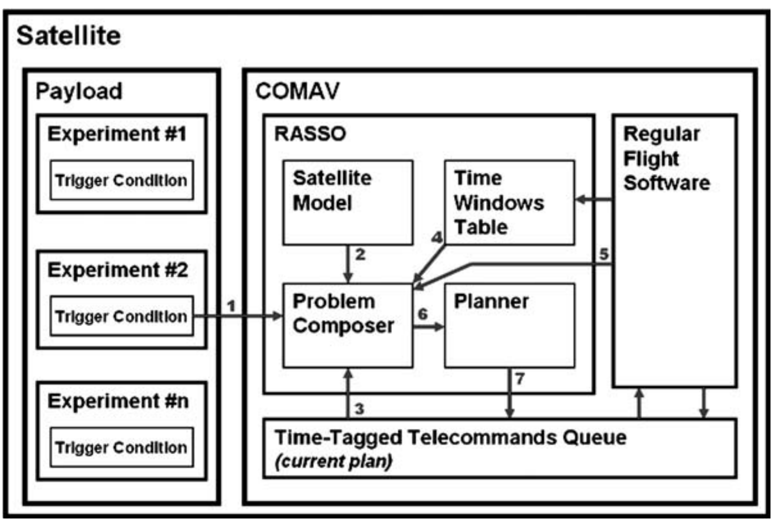


Fig. 1 RASSO architecture.

observation of the phenomenon (arrow number 1, to the left). When receiving the request, RASSO composes a well-defined problem in the form of a draft operation plan. This draft plan is created consolidating information from several sources inside the onboard software (arrows 2–5), and then is directed to the planner module (arrow 6), responsible for working out the conflicts that were inserted in the operation plan because of the request from the experiment, respecting a set of constraints and goals that were imposed on it. When succeeding in creating a new operation plan that takes care of all of these requirements, RASSO sends it to the Time-Tagged Telecommands (TTTC) queue, turning it in the new plan of the satellite's experiments (arrow 7). RASSO onboard service is explained in more detail next.

IV. EQUARS Satellite Model

As previously stated, our case of study is the EQUARS scientific satellite. EQUARS is the next satellite of INPE's scientific satellites series. It is a project in partnership with research institutions from many countries. The embarked experiments are proceeding from institutions from Brazil, United States, Canada, and Japan, and have as its aim to make a global scale monitoring of the Earth's equatorial low, middle, and upper atmosphere and ionosphere, with a special emphasis in dynamical and photochemical energy transport processes. EQUARS is a project in course, and the set of expected scientific experiments for the satellite is the following:

1) IONEX, a plasma sensor for the measurement of the plasma density and electronic temperature.

2) GROM, an instrument based on the reception of the GPS constellation signals for the measurement of humidity, temperature, and total electron content.

3) CERTO, a beacon transmitter that will make observations of ionospheric irregularities, electron content, and scintillations.

4) TIP, a luminescence imager for lightning and sprites.

5) MLTM, a mesopause temperature imager.

It is necessary to model the satellite components and dynamics to allow RASSO to infer its behavior. However, the way a system is modeled has a direct impact in the planner performance. AI planning is generally associated with a huge processing volume and memory consumption. This does not present a great problem when working with PC computers capable of running with clock frequencies in the order of gigahertz and have several hundreds of megabytes of RAM memory. Nevertheless, this is exactly the opposite of what is found in embedded systems.

COMAV is based on an ERC32 processor (RISC, SPARC 32 bits architecture) running at 12 MHz, and has 2 Mb of program memory. The computational power limitation and the quickness necessary to reconfigure the satellite in response to a phenomenon force us to take special care with the planner performance—the total replanning time has to be kept in the order of some few minutes. In these conditions, the time spent to analyze a model described in a language such as PDDL [17] at run time, and to transfer its elements to adequate data structures, starts to be significant.

To deal with these limitations, we decided that the satellite model to be used by the planner should be described in the proper C language. The model elements

would be stored directly in the data structures in which they would be worked, preventing the model's description analysis stage.

The C language, if used in the right way, can describe a model adequately, but a model described in C would lose in clarity. An engineer or scientist not familiar with the programming language and with the software structure would not understand what is being represented. This way, to allow the direct storage in the right data structures and still keep the model readable, we decided to create a model description language over C, through the use of macros, which hide all of the structures, pointers, and function calls used by the planner.

The use of macros to implement this language—that we called RASSO_ml—makes the task of converting a model instruction to data structures the responsibility of the GCC compiler's pre-processor. With this, the model elements are always ready to be used by the planner, eliminating a great part of the initialization process.

Of course there are pros and cons to this approach. Citing some downsides, sometimes the RASSO_ml features are implemented in a not-so-elegant way. For example, it is necessary to inform the number of objects to be instantiated when creating a class. Moreover, compilation error messages referring to RASSO_ml instructions can be not clear, since they are related, in fact, to the C structures and not to the instructions to the planner. In favor, besides the clarity of the model description and the reduction in the model's initialization time, already cited, we can point out that the use of RASSO_ml elements alongside with the C applications source code allow a more natural integration of the planner with the rest of the software.

The satellite model is edited in conjunction with the COMAV's software source code. It must be in a model.h file, which is linked to the RASSO_ml library and to the planner. A model represented in RASSO_ml is composed by a static description (structural) and a dynamic description (behavioral), which are described in detail as follows.

A. Model Static Description

The static part of the model is composed basically of objects and resources. Objects are elements instantiated from classes defined by the modeler, which have attributes whose values are changed by the commands execution or the occurrence of exogenous events. The set of attribute values of an object at a specific moment in time is generally called object state. Resources are the consumable elements of the model.

There are three kinds of resources in RASSO_ml: exclusive, depletable, and reservable. The exclusive resources do not have a quantity (they are unique) and can be used by only one object at a time. The depletable and reservable ones have a minimum and maximum quantities defined in its creation, and these quantities cannot be exceeded in any moment. The difference between them is that while the depletable resources "accumulate" its consumption in time until being completely depleted, the reservable ones are controlled in a momentary way, without accumulation. Examples of depletable resources are fuel and memory; power, in its turn, is a reservable resource. Resources are controlled in units, with no matter to the planner if they are watts, kilos, or bytes. Figure 2 shows a simplified RASSO_ml code snippet with instructions to the planner for the creation of a domain, a class, objects, and resources. The objects initialization code is not shown in the figure.

```
create_domain(Exp_Name, ionex, grom, certo, tip, mltm);

create_class(Experiment, 5,
                         Exp_Name      name;
                         Boolean       on;
                         int           sample_rate;
                         int           priority;
             );

create_object(IONEX, Experiment);
create_object(MLTM,  Experiment);

create_resource(Power, reservable, 0, 60);
create_resource(Mass_Memory, depletable, 0, 128);
```

Fig. 2 Creating classes, objects, and resources.

B. Dealing with Time

An operation plan is a set of actions (satellite's internal commands) that affects the satellite state as the time goes by. (The term *command* is reserved for the satellite software commands, while *instruction* refers to instructions to the planner.) For the planner to be successful in dealing with the changes of the modeled satellite, it has to manage not only the objects and resources, but also all of the states they assume over the plan, that is, all of its "moments." Thus, whenever an object is created, it is creating, in fact, a timeline for the object, which stores all of the states it assumes during the plan period. An object is not directly manipulated in RASSO_ml; it is necessary to declare in which moment of the object one wants to work. Figure 3 shows some of the ways of dealing with objects in time.

In a similar way of what happens with the objects, when creating a resource, a resource consumption profile is generated. This profile controls how much of each resource is being consumed or generated at each moment of the time and is used to detect resource overuse.

C. Model Dynamic Description

The main elements of the dynamic part of the model are the actions and the behaviors. An action corresponds to one or more internal satellite commands. When composing a problem, RASSO converts commands extracted from the TTTC queue in its corresponding actions to generate an initial draft plan. When the planning process is finished, the planner sends to the TTTC queue the

```
current_state(TIP).on = false;

if (initial_state(TIP).on == goal_state(TIP).on) return(success);
```

Fig. 3 Working with timelines.

commands corresponding to the new plan that overlaps the old one (this will be explained in more detail in the next sections). The action is implemented as a C language function and describes how the model is affected by its execution. The left panel of Fig. 4 shows a simple action, which is described as follows.

An object of the type Experiment is sent to the Turn_On action. Each action is called by the planner at two different moments in the planning process. The blocks when_planning and when_running indicate which code fragment must be executed in each one of these moments. The first moment (when_planning) happens when the planner is testing actions to apply in an incomplete plan, trying to achieve the goals and satisfy the constraints that were imposed on it. The second moment (when_running) happens when the plan was already obtained, and the planner is sending the commands corresponding to the actions to the TTTC queue. The function Send_To_TTTC_Queue called here is an internal satellite pseudocommand. The when_planning block holds the preconditions necessary for the execution of an action and its effects over the model in case all of the conditions are true.

```
RASSO_Action Turn_On (with_parameters)
{
  parameter(exp); // experiment

   when_planning
   {
      // the gps cannot be turned on / off !!!
      condition(current_state(exp).name != grom);
      condition(current_state(exp).on == false);

      // effects described here
      current_state(exp).on = true;

      switch(exp.name)
      case ionex:
      {
          consumes(exp, Power, 7 per_hour);
          consumes(exp, Mass_Memory, 1.5 per_min);
          break;
      }
      case mltm:
      {
          consumes(exp, Power, 4 per_hour);
          consumes(exp, Mass_Memory, 0.8 per_min);
          break;
      }
  }

  when_running
  {
      Send_to_TTTC_Queue(Turn_On_Exp(exp));
  }

  action_success;
}
```

```
RASSO_Behavior Day_Bhv (happens_at Day)
{
   at_start
   {
       generates(Power, 0.5 per_min); // load battery

       // "don't mess with MLTM memory!!!"
       keep_resource_untouched(MLTM, Mass_Memory);
   }
   at_end
   {
       generates(Power, 0); // stop loading battery
   }
}
RASSO_Behavior Night_Bhv (happens_at Night)
{
  at_start // sun sensor turns on experiments
  {
      Turn_On(IONEX);
      Turn_On(CERTO);

      // guarantee at least 4 Watts for 1/2 hour to IONEX
      guarantee_resource(IONEX, Power, 4, 0, 1800);
  }
  at_end // sun sensor turns off experiments
  {
      Turn_Off(IONEX);
      Turn_Off(CERTO);
  }
}
RASSO_Behavior Comm_Bhv (happens_at Comm)
{
  at_start
  {
     // send data and frees memory at 230kbps
     generates(Mass_Memory, 13.8 per_min);
  }
  at_end
  {
     generates(Mass_Memory, 0); // stop sending data
  }
}
```

Fig. 4 Actions and behaviors in RASSO_ml.

The "consumes" instruction deserves a more detailed analysis. It informs that, from the execution of the action on, the experiment passed as parameter will consume the power resource at the rate informed in the instruction—seven units of power per hour. The basic RASSO time unit is the second. When informed of the consumption rate over time, the planner stores it in seconds. The macros per_min, per_hour, and per_day are just multipliers.

Actions allow describing the satellite behavior in response to determined commands executed by software. However, not all of the changes in the satellite state happen as a function of software commands. Some of the EQUARS experiments have their functioning tied to the solar incidence, or the lack of it. The TIP experiment, for instance, makes observations of lightning and sprites while it is in eclipse (at night, during an orbit), what is not possible when it is illuminated (at day). These experiments can be activated and deactivated automatically by solar sensors, without turn on/off commands scheduled for execution. Although the software is notified that the experiment was turned on/off, there is no way to predict this with antecedence, which is vital for the planning process. To deal with exogenous events such as the ones just described, RASSO makes use of the concept of time windows.

Time windows are periods in which the satellite presents a determined typical behavior. For example, during the period in which the satellite is in contact with a ground control station, it transmits the experiments stored data, thus releasing the resource memory. It also consumes more power since its communication system is active. Figure 5 shows the creation of three time windows: Day, Night, and Communicating. The second instruction parameter is a unique time window identifier, used by the planner.

Time windows can be consecutive, as is the case of Day and Night, or they can overlap each other, as Communicating in relation to the other two—no restriction about this is made by RASSO. There is also no obligation concerning the occurrence of a time window in every orbit. The window Communicating will not occur in every orbit, depending on the orbital characteristics. RASSO has a time windows table, which stores the start and end times of the occurrence of each time window for every orbit in the following days. These data are updated regularly by the ground operations team via telecommands. Using this information, it is possible to tie behaviors to the occurrence of the windows.

The behavior is part of the model dynamic description. While an action must have one or more commands related to it in the TTTC queue, the behavior describes activities that happen at the beginning and/or the end of a time window, independently of the current operation plan. These activities can be described

```
// time windows related to orbital phases
create_time_window(Day, 0);
create_time_window(Night, 1);
create_time_window(Communicating, 2);
```

Fig. 5 Creating time windows.

directly in the behavior, through instructions for the manipulation of objects and resources, or through calls to actions—in this case, only the block "when_planning" of the action is executed. Behaviors, in contrast to actions, do not generate commands in the plan generated by RASSO (see the right panel of Fig. 4, which gives behavior examples).

The "happens_at" clausule ties a behavior to a time window. The blocks "at_start" and "at_end" of each behavior indicate which activities are related to the beginning and end of the time window. In the Fig. 4 example, the instruction generates (Power, 0.5 per_min) indicates that the battery is being loaded at the rate of 0.5 units per minute while the satellite is enlightened. At the end of the time window, generates(Power, 0) informs that the loading is finished. The instructions "keep_resource_untouched" and "guarantee_resource" can only be used in the "at_start" block of the behavior and impose constraints to the planner. The first one prevents the planner from selecting any action that affects the resource consumption by an object. In the example, it prevents changes in the consumption of memory by the MLTM experiment. The second one imposes that a determined amount of resource must be guaranteed to an object during a certain period inside the time window (in the example, it is imposed the maintenance of at least 4 units of power for the IONEX experiment, on the first 30 min of the Night window).

V. Problem Composer Module

COMAV will communicate with intelligent experiments, that is, experiments that are managed by their own processor and software. Each experiment that can detect short-duration phenomena shall have at least one trigger condition in its software that, when true, will send a request for more resources to RASSO. The request notifies which resources are needed, how much is necessary, from what moment, and for how much time. This request for resources will be received by the problem composer module, which is responsible for congregating information from several sources (see Fig. 1) and supply the planner with a problem to be solved.

The creation of a well-defined problem is as important as the process of searching for the solution. The problem consists of a draft operation plan with goals to be achieved, constraints to be respected, and conflicts to be solved. The initial satellite state, the goal states, the telecommands (actions) scheduled for execution, and the exogenous events (behaviors) that occur during the plan execution period are all part of the problem.

A. Planning Horizons

The request for resources sent by the experiment carries in its parameters the information of two crucial moments for the planning process: the beginning and end of the period in which the experiment needs more resources. These key moments are called planning horizons, and they are used to determine the initial state, the intermediate goals, and the final goal (see Fig. 6).

The beginning moment of the period with more resources allocated for the solicitant experiment is called horizon h1, or simply h1. The end moment of the period with more resources is h2. The horizons h1 and h2 indicate the moments in which the intermediate goal states are in the planning process.

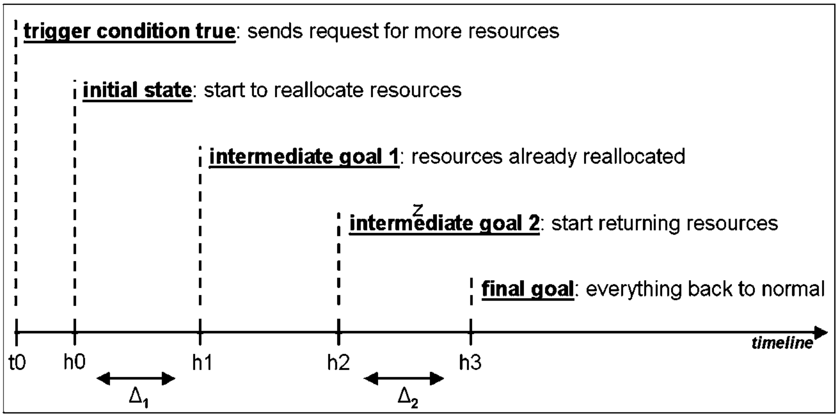


Fig. 6 Planning horizons and goals.

The horizon h1 indicates the moment in which the requested amount of resources must be already reallocated for the experiment; h2 indicates until when these resources must be kept. In function of these, there are two more horizons: the initial horizon h0 and the final horizon h3; h0 is determined as being h1 – Δ1, where Δ1 is the time necessary for the execution of reallocation commands for the solicitant experiment. In a similar way, h3 is determined as h2 + Δ2, where Δ2 is the time necessary to return the resources to the experiments that yielded them, thus placing the satellite in its "normal" operation mode.

It is necessary to define the meaning of normal operation mode here: as RASSO is proposed to temporarily modify the satellite's operation mode, the plan generated shall guarantee that, when reaching the horizon h3, all of the modeled objects will be in the same state as they would if the original plan were executed, and the available amount of resources will be at least equal to that that would be left by the execution of the original plan.

B. Composing a Well-Defined Plan

The problem composition goes through three main stages: the attainment of the model initial state in horizon h0, the attainment of the current operation plan between h0 and h3, and the imposition of constraints and intermediate and final goal states to the planner, including the conflicts to be solved. These stages are summarized as follows:

1) The model is initialized and the current state of objects and resources are obtained through calls to other application processes.

2) The current operation plan is read from the TTTC queue, and its time-tagged commands are applied to the model, from the first on the queue to the one immediately before the horizon h0.

3) The time windows table is read. It is verified what windows initiate or finish until h0. The behaviors tied to these time windows are applied to the model,

respecting the at_start and at_end blocks, from the current moment until h0. By doing this, the initial state is reached.

4) Steps 2 and 3 are repeated, now between the horizons h0 and h3. By doing this the problem composer gets the effects of the execution of the current operation plan over the satellite, and can start imposing constraints to the planner.

5) The satellite state at the end moment of the plan execution period (horizon h3) is marked as "final goal." This makes the planner respect this state and search for actions that take the satellite to them during the planning process.

6) Applied to the draft plan is the request for more resources sent by the experiment. This is made imposing a constraint to the planner that it has to keep at least the amount of resources requested by the experiment during the period between h1 and h2.

7) The guarantee_resource and keep_resource_untouched instructions are applied, imposing the last constraints to the planner.

When finishing these stages, the problem composer directs to the planner module a draft operation plan with goals to achieve, conflicts to solve, and constraints to respect.

VI. Planner Module

Having a draft plan, the planner module will cover it in chronological order from the horizon h0 to h3, searching for conflicts to solve. Whenever a conflict is found, it will retrocede in the plan and try to change the actions in such a way to resolve it. Once the conflict is resolved, the planner returns to scroll the plan in direction to h3, searching for the next conflict to solve.

Conflicts can be of two kinds: violation in the resources consumption constraints or inconsistent goal states. A violation conflict in the resources consumption constraints is related to the resource overuse, consuming more than its available amount, or a quantity out of the range imposed by the problem composer. To resolve this kind of conflict, the planner has at its disposal the following options:

1) Change the operation mode of experiments to make them consume less. To do this, it is possible to try an action Change_Sample_Rate, for instance, to modify the data acquisition interval.

2) Delay the execution of an instruction to turn on an experiment—since this instruction is an action (a time-tagged telecommand) and not a behavior (an exogenous event).

3) Turn off an experiment for the least possible time.

An inconsistent state conflict is related to goal states imposed on the planner by the final goals in h3 or by instructions inside behaviors. When the execution of actions and behaviors does not lead to one of the imposed states, there is a conflict to solve. The planner tries then to find actions that take the model objects to the desired states. In both cases, heuristics guide the search process to choose the action that seems to lead the satellite to a state nearer the goal. However, the insertion, alteration, or exclusion of actions in the plan can insert new conflicts, which are treated by the planner in the same way. This process continues until there are no more conflicts to solve. This approach is named in DCAPS and its successors as *iterative planning*.

When the plan does not have more conflicts, it is sent to the TTTC queue, to become the new satellite's operation plan, overlapping the old one.

VII. Onboard Replanning in a Safe Way

It is a fact that there is a great resistance by engineers and mission managers to increase satellite autonomy. The cost and amount of work of a space mission is, in general, considered big enough to trust the satellite to make decisions that are more than routine, such as the entrance in the emergency mode, or the attitude and orbit correction.

This resistance is even bigger when the increase of autonomy is propitiated by a technique coming from the artificial intelligence area. Even projects that have the intention of being a test bed of technologies to increase the satellite's autonomy, such as ESA's PROBA and others missions in which requirements are perfect for the use of planning, such as the SWIFT observatory from NASA [18], do not implement any AI technique onboard. Thus, how to overcome the resistance and convince a mission manager to use onboard planning in his satellite? To get RASSO validated in the EQUARS satellite, we are following some rules, described as follows.

First, the planner must propose to solve a small problem, and not one that is vital to the satellite functioning. The reallocation of resources propitiated by RASSO is desirable, but if it is not possible, the experiments keep collecting and processing data in the normal mode of operation. Second, the service implementation must be gradual and based on the increase of confidence with respect to the results obtained. RASSO has three distinct modes of operation that guarantee this: disabled, advice, and act.

In the disabled mode the service is not available, and the experiments will always have their requests ignored. The advice mode exists to allow the analysis of the service reliability. In this mode, the service works in a plan as it would really meet the request, but with a reduced priority during the planning process. The resulting plan is not sent to the TTTC queue but is made available to telemetry, alongside with a service activity log. Every time that a request is sent to RASSO, when in advice mode, an alert message showing that there are plans stored in the satellite waiting for analysis is sent through normal telemetry, no matter if the obtained plan is complete or not—that is, if the planner was or was not successful in simulating a satisfactory response to the request.

The generated plan is sent to the ground in reply to a telecommand specific for this purpose. The actions generated by the planner are then analyzed by the satellite operators and compared through proper tools with the normal satellite plan, which was really executed. If it is determined that RASSO has generated a plan that would meet the request of an experiment, still keeping the normal operation of the satellite, this is computed as a success of the service. On the contrary, the planning algorithm and the satellite model have to be reviewed, and a new version is sent to the satellite. Having a number of successes considered enough, the service can be set to the act mode and start to be completely operational. Then, the generated plans are sent directly to the TTTC queue.

In addition to the approach just described, it must also be considered that the target application of RASSO is the scientific satellites. These are missions of lower cost, where there is more room for experimentation.

VIII. Conclusion

Given the level of development reached by onboard satellite systems, the next step to take is the increase of its autonomy. Providing technology for the use of onboard planners meets that, but it is something to be treated carefully because of the criticality of the application. To accomplish this, the project proposes, allied to the technology, the use of a gradual process, based on the increasing of confidence of the results obtained. It is intended with that to open new possibilities to be exploited for the INPE's technological and scientific experiments aboard satellites.

Through modest goals and a realistic approach, RASSO consists of a first step toward the use of onboard planning to increase the autonomy of our satellites in the near future.

References

[1] Fukunaga, A., Rabideau, G., Chien, S., and Yan, D., "Towards an Application Framework for Automated Planning and Scheduling," *Proceedings of the International Symposium on Artificial Intelligence Robotics and Automation in Space*, IEEE Press, Tokyo, July 1997, pp. 375–386.

[2] Zweben, M., Daun, B., Davis, E., and Deale, M., "Scheduling and Rescheduling with Iterative Repair," *Intelligent Scheduling*, edited by M. Fox and M. Zweben, Morgan Kaufmann, San Mateo, CA, 1994, pp. 241–256.

[3] Myers, K. L., and Smith, S. F., "Issues in the Integration of Planning and Scheduling for Enterprise Control," *Proceedings of the DARPA-JFACC Symposium on Advances in Enterprise Control*, San Diego, CA, Nov. 1999, p. 1.

[4] Fikes, R. E., and Nilsson, N. J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem-Solving," *Artificial Intelligence,* Vol. 2 (Nos. 3–4), 1971, pp. 189–208.

[5] Johnston, M. D., and Miler, G. E., "SPIKE: Intelligent Scheduling of Hubble Space Telescope Observations," *Intelligent Scheduling*, edited by M. Fox and M. Zweben, Morgan Kaufmann, San Mateo, CA, 1994, pp. 391–422.

[6] Chien, S., Smith, B., Rabideau, G., Muscetolla, N., and Rajan, K., "Automated Planning and Scheduling for Goal-Based Autonomous Spacecraft," *IEEE Intelligent Systems*, Vol. 13, No. 5, 1998, pp. 50–55.

[7] Sherwood, R., Govindjee, A., Yan, D., Rabideau, G., Chien, S., and Fukunaga, A., "Using ASPEN to Automate EO-1 Activity Planning," *Proceedings of the IEEE Aerospace Conference*, Aspen, CO, April 1998, pp. 145–152.

[8] Fuchs, J. J., Gasquet, A., Olalainty, B., and Currie, K., "PlanERS-1: An Expert Planning System for Generating Spacecraft Missions," *International Conference on Expert Planning Systems*, Brighton, England, UK, 1990, pp. 70–75.

[9] Aarup, M., Arentoft, M. M., Parrod, Y., Stokes, I., Vadon, H., and Stager, J., "Optimum-AIV: A Knowledge-Based Planning and Scheduling System for Spacecraft AIV," *Intelligent Scheduling*, edited by M. Fox and M. Zweben, Morgan Kaufmann, San Mateo, CA, 1994, pp. 451–469.

[10] Cesta, A., Oddi, A., Cortellessa, G., Fratini, S., and Policella, N., "AI Based Tools for Continuous Support to Mission Planning," *SpaceOps 2006 Conference*, Rome, Italy, June 2006.

[11] Bernard, D., Dorais, G., Gamble, E., Kanefsky, B., Kurien, J., Man, G. K., Millar, W., Muscettola, N., Nayak, P., Rajan, K., Rouquette, N., Smith, B., Taylor, W., and

Tung, Y. W., "Spacecraft Autonomy Flight Experience: The DS1 Remote Agent Experiment," *Proceedings of the AIAA Space Technology Conference*, Albuquerque, NM, Sept. 1999; AIAA Paper 1999-4512.

[12] Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castaño, R., Davies, A., Mandl, D., Frye, S., Trout, B., D'Agostino, J., Shulman, S., Boyer, D., Hayden, S., Sweet, A., and Christa, S., "Lessons Learned from Autonomous Sciencecraft Experiment," *Autonomous Agents and Multi-Agent Systems Conference*, Utrecht, The Netherlands, July 2005, pp. 11–18.

[13] Teston, F., Creasey, R., Bermyn, J., Bernaerts, D., and Mellab, K., "PROBA: ESA's Autonomy and Technology Demonstration Mission," *Proceedings of the 13th AIAA/USU Conference on Small Satellites*, Logan, UT, Sept. 23–26, 1999.

[14] Cardoso, L. S., Ferreira, M. G. V., and Orlando, V., "An Intelligent System for Generation of Automatic Flight Operation Plans for the Satellite Control Activities at INPE," *SpaceOps 2006 Conference*, Rome, June 2006.

[15] Biancho, A. C., Carniello, A., Ferreira, M. G. V., Silva, J. D. S., and Cardoso, L. S., "Multi-Agent Ground-Operations Automation Architecture," *Proceedings of the 56th International Astronautical Congress*, Fukuoka, Japan, Oct. 2005.

[16] Kucinskis, F. N., and Ferreira, M. G. V., "Dynamic Allocation of Resources to Improve Scientific Return with Onboard Automated Planning," *SpaceOps 2006 Conference*, Rome, June 2006.

[17] Edelkamp, S., and Hoffmann, J., "PDDL 2.2: The Language for the Classical Part of the 4th International Planning Competition," Albert Ludwigs Universität Institüt fur Informatik, Technical Rept. 195, Freiburg, Germany, Jan. 2004.

[18] Ong, J. C., and Rackley, M., "Autonomous Operations for the SWIFT Mission," *Proceedings of the Seventh International Conference on Space Operations*, Houston, TX, Oct. 2002.

This page intentionally left blank

VI. Planning Tools and Advanced Technologies

This page intentionally left blank

Chapter 21

In-Space Crew-Collaborative Task Scheduling

John Jaap,* Patrick Meyer,* Elizabeth Davis,* and Lea Richardson*
NASA Marshall Space Flight Center, Huntsville, Alabama

I. Introduction

FOR all past and current human space missions, the final scheduling of tasks to be done in space has been devoid of crew control, flexibility, and insight. Ground controllers, with minimal input from the crew, schedule the tasks and uplink the timeline to the crew or uplink the command sequences to the hardware. Prior to the International Space Station (ISS), the crew could make requests about tomorrow's timeline, they could omit a task, or they could request that something in the timeline be delayed. This lack of control over one's own schedule has had negative consequences [1]. There is anecdotal consensus among astronauts that control over their own schedules will mitigate the stresses of long-duration missions. On ISS, a modicum of crew control is provided by the "job jar." Ground controllers prepare a task list (also known as the job jar) of non-conflicting tasks from which jobs can be chosen by the in-space crew. Because there is little free time and few interesting non-conflicting activities, the task-list approach provides little relief from the tedium of being micromanaged by the timeline.

Scheduling for space missions is a complex and laborious undertaking that usually requires a large cadre of trained specialists and suites of complex software tools. It is a giant leap from today's ground-prepared timeline (with a job jar) to full crew control of the timeline. However, technological advances, currently in-work or proposed, make it reasonable to consider scheduling a collaborative effort by the ground-based teams and the in-space crew. Collaboration would allow the crew to make minor adjustments, add tasks according to their preferences,

*Computer Engineer, Data Systems.

understand the reasons for the placement of tasks on the timeline, and provide them a sense of control. In foreseeable but extraordinary situations, such as a quick response to anomalies and extended or unexpected loss of signal, the crew should have the autonomous ability to make appropriate modifications to the timeline, extend the timeline, or even start over with a new timeline.

The Vision for Space Exploration (VSE), currently being pursued by NASA, will send humans to Mars in a few decades. Stresses on the human mind will be exacerbated by the longer durations and greater distances, and it will be imperative to implement stress-reducing innovations such as giving the crew control of their daily activities.

A. Major Consideration

Implementation of crew collaboration will be driven by one major circumstance—the round-trip light-time delay between Earth and Mars can be up to 44 min and will never be less than 6 min (Fig. 1). Every 26 months, Mars cycles between close approach and far retreat. Barring some breakthrough in propulsion technologies, a mission to Mars will take longer than two years, during which the light-time delays will average over 20 min. Normal voice conversations will be impossible.

Instant transfer of information as provided by today's Earth-based Internet will also be negated. The communication delays will change the way human missions are operated. The crew will be the first responders to emergencies and mundane anomalies; they will attend autonomously to all alarms, switching the troublesome system to a safe mode and/or making quick repairs and reconfigurations. What we now think of as ground *control* teams will become ground *support* teams.

The inability to have normal conversations with the ground, seeing the Earth only as a point of light, having no immediate return-to-Earth options, interacting only with their companions, having limited food choices, drinking recycled water, and doing the same maintenance tasks each day will make life stressful for the crew. Crew collaboration on the development of the daily timeline for themselves and for the systems they use and maintain will provide them necessary knowledge to execute the timeline and necessary influence so that they will feel that they are in control of their own destiny.

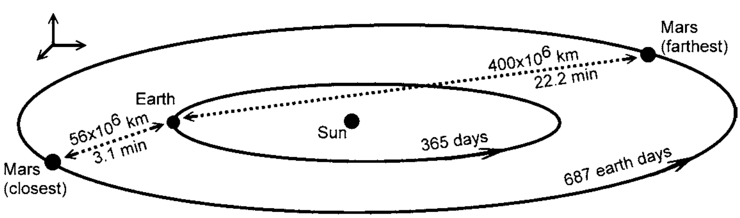


Fig. 1 Light-time delays to Mars.

B. Collaboration

Collaboration is working jointly to produce a product or to attain a goal. Usually collaborators share ideas, compromise on goals or methods, contribute where their expertise allows, and jointly arrive at an objective. There are two types of collaboration, interactive and passive. Interactive implies that the collaborators communicate during the collaboration process. Passive collaboration happens without direct communication. An instance of interactive is a group of authors who brainstorm the contents of a paper. An instance of passive collaboration is the progression of teachers who instruct children from early childhood into adults; another instance is the corps of guards who attend all of the gates at an arena so that only ticket holders are allowed to enter.

While passive collaboration is based on a concept of operations rather than communication, there are several methods that support interactive collaboration; often, these are used in combinations. The more common methods are listed in approximate order of productivity as follows.

1) Face-to-face: collaborators talk to one another across the table, use whiteboards, share notes, etc.

2) Teleconferencing and web conferencing: collaborators simulate being face-to-face using voice, possibly video, and sometimes an electronic white board or a shared computer "desktop."

3) Custom software: an application designed to support collaboration on a specific task or product. Internet games are a common example of custom software.

4) Instant messaging and chat rooms: collaborators type messages that are displayed almost instantly for each collaborator.

5) File transfer: collaborators post and retrieve files using peer-to-peer transfer or a common drop box. Commercial products are available to automate this method.

6) Electronic forums: collaborators post messages on a message board.

7) Electronic mail: collaborators correspond by sending messages over the Internet.

8) Postal mail: collaborators correspond by written mail.

For Earth-based support teams and humans on a mission to Mars, several of these collaboration methods will not be available and others (file transfer, electronic forums, and electronic mail) will be degraded. Custom software and the concept of operations can be tailored to deal with the time delays. Collaboration between the in-space crew located at Mars and the ground support will, no doubt, use all of the tools available, including a delay-tolerant communication infrastructure, delay-tolerant scheduling software, and a specially tailored concept of operations.

C. Integration

In any large community of collaborators, terminology and interfaces are important. The VSE will be the first time that intelligent robots and rovers have been on the same mission as humans. A master timeline will be needed that integrates the timelines of the crew, automatic systems, and intelligent systems. Any member of the community of contributors may need to view or change any timeline. For example, the crew may need or want to schedule the activities of the rovers. The scheduling requirements for all systems, including the crew, must be described using the same terminology, likewise for the resulting timelines.

The user interfaces to the software must be similar for all collaborators—the timeline for a crewmember and the timeline for an automatic system must be viewable on the same display. Additionally, the interface must have special instances for use by the crew in the cabin of the habitat or in their spacesuit while walking on the surface of Mars.

II. Concept of Operations

Human missions to the moon and Mars will use extraordinarily complex hardware and software systems and will include significant technological and scientific investigations. The missions will extend for years, and ground support will require collaboration from many widely dispersed contributors. The cost to collect the ground-based planning and scheduling collaborators in a central location will be prohibitive. The only affordable solution will be to distribute the planning and scheduling efforts. It logically follows that if the planning and scheduling effort is to be distributed, it can be distributed to the crew on the moon, in transit to Mars, and at Mars. Distributed planning and scheduling has been used to a small extent for ISS. Research into concepts of operations [2] that maximize the distribution and the cost savings has yielded viable results.* To be successfully accepted and implemented, a distributed concept of operations must be considered and developed beginning as early as possible. The concept proposed here is intended to be a starting point for the concept that will eventually be used to allow full distribution of the planning and scheduling function to all collaborators including the in-space crew. The concept has the additional advantage of giving the crew full autonomy if they should need it.

A. Enabling Principles

The concept of operations is based on several principles that enable collaboration:

1) Adding or removing items from the timeline by one collaborator does not invalidate the contributions of another collaborator.

2) Collaborators need only minimum expertise in the knowledge realm of other collaborators.

These principles apply to the crew as they modify the timeline. The crew is often tempted to make on-the-fly modification to the timeline as they execute it (sometimes we say that the crew interprets the timeline); these principles require that they allow the system to record and verify proposed changes to the timeline.

B. Overview

Figure 2 shows the contributions to the planning and scheduling effort by different collaborators during the preparation of the preliminary timeline on Earth and

*Fully distributed concepts of operations will not be implemented for ISS because the paradigm shift would require rewriting current international agreements; incur appreciable one-time costs, including new software, procedures, and training; and the phaseover would extend almost to the end of the ISS mission.

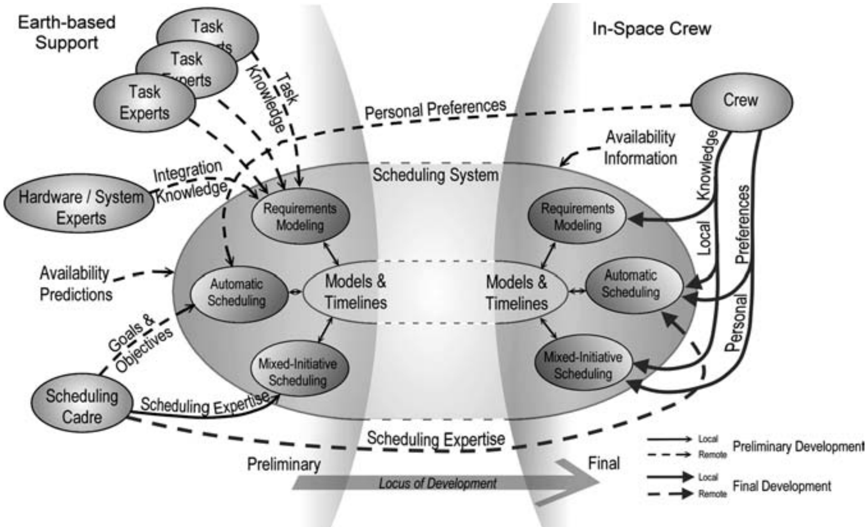


Fig. 2 Concept of operations.

after the timeline is uplinked to the crew. Text messages, notes, e-mail, and voice memos are not shown on the chart. The development of the timeline is divided into two phases, preliminary and final. Preliminary timelines are developed on the ground using an Earth-based instance of the scheduling system. Before the beginning of each work day, the timeline is transferred to an in-space instance of the scheduling system. Crew members have time-delayed access to the ground instance and the ground support has time-delayed access to the in-space instance.

C. Widespread Collaboration on Preliminary Timeline

Collaboration is cost effective because those with the best knowledge are able to contribute that knowledge directly. For the same reason, collaboration produces the best product. Collaboration on the planning and scheduling effort for the preliminary timeline is described by listing what each collaborator contributes as follows:

1) Task experts: Task experts include hardware developers, scientists, engineering support teams, doctors, astrobiologists, and others who have first-hand knowledge about the tasks to be done to maintain the vehicle/habitats and to conduct the exploration and science activities. These experts define the tasks and arrange them into the required sequences to accomplish the goals. The task definitions include the procedure descriptions, the equipment (and operation modes) to be used, the conditions required, and the durations. These task models do not directly define the resource amounts required by the tasks; resources are defined by the equipment mode models entered by the hardware and systems experts. The sequence definitions include the temporal relationships between the tasks, relationships to other sequences, windows, of opportunity, and other data. The information is entered into a central data repository used by the scheduling system.

2) Hardware and system experts: Hardware and systems experts have detailed knowledge about how the hardware is integrated into the vehicle(s) and how that hardware can be used. They also have knowledge about the software systems and how they behave. The experts enter the equipment mode models into the central data repository. For each mode of each piece of equipment, these models define how much of each resource is used. For example, an oven for metallurgical analysis of regolith samples has several operation modes using different resources (power, inert gases, data rates, etc.) and amounts. The hardware and systems experts know how the oven functions and how it is connected to the systems (which power bus, which controller, which video bus, etc.). These equipment mode models are used by the task experts when defining the tasks; therefore, these models eliminate the need for task experts to be hardware experts.

3) Scheduling cadre: The scheduling cadre interacts with program managers, engineering support teams, and the science teams to determine the day-by-day plans. Based on the plan, the cadre submits task sequences to the scheduling engine, which adds them to the timeline. The scheduling engine accepts remote requests into a queue of sequences to be scheduled. For each sequence, the engine combines the task models with the equipment mode models to create a complete model including all timing, resources, relationships, and conditions, and it then schedules the request. The scheduling cadre can use a timeline editor (mixed-initiative scheduling) to make final tweaks to the timeline. The modeling and scheduling engine are powerful enough that mixed-initiative scheduling will be required only in rare circumstances.

4) Crew: Crew members can send messages, receive messages, and read the minutes of the various planning group meetings. The crew has first-hand knowledge of the local situation. The crew can collaborate on the development of the preliminary timeline because they can view the task models, the equipment mode models, the developing (partial) timeline, and *the crew can submit scheduling requests* to the instance of the scheduling engine currently being used by the ground support teams.

5) Other contributors: Using appropriate software, other contributors provide availability predictions for resources (such as power), conditions (such as daylight or communications), and companion autonomous systems (such as rovers or robots).

In addition to developing the preliminary timeline, the collaborators also develop a task list containing sequences that are candidates for adding to the timeline. These may be purely discretionary or they may be sequences that are planned for the near future.

D. Collaboration on Final Timeline

The final timeline is developed in space. On the day or evening before execution, the preliminary timeline and all supporting data are transferred to an instance of the scheduling system that is co-located with the crew. Crew members have immediate and full access to all features of the system, and they are the main contributors to the finalization of the timeline. They have first-hand knowledge of the in-space systems; they know their own preferences and needs. They can remove omitted tasks from the timeline so that unused resources are known by the system to be available. They can modify the equipment mode models to reflect actual in-situ

configurations, and they can modify task and sequence models to reflect personal experience. They can add to the timeline by selecting items from the proposed task list and submitting them to the scheduling engine so that the tasks are assigned times and so that the resource usage is tracked.* The modification to the timeline includes not only tasks done by the crew but also unattended tasks done by automated or autonomous systems such as robots. In rare circumstances the crew could use mixed-initiative scheduling to modify the timeline; however, mixed-initiative scheduling can consume a lot of valuable crew time, and a system that relies on extensive mixed-initiative scheduling will not be acceptable.

Ground support teams collaborate on the final adjustments to the timeline by reviewing modifications to the models or by updating the models (updating the Earth-based instance and uplinking it). They collaborate on timeline additions by reviewing the crew's modifications and commenting via text messaging. The ground teams can also use remote access to submit new requests to the in-space scheduling engine.

E. Crew Autonomy

When a situation arises in space that necessitates modification and execution of the timeline before the ground can see the change (due to light-time delays), the crew can go ahead and make the change. It is essential for the crew and the in-space vehicle or habitat to be able to function if there is an extended communication loss. The crew may need to extend the timeline for a few hours or for many days. The needed autonomy can be provided by installing in space a complete planning and scheduling system that is sufficiently automated so that the crew can build a timeline with minimum effort.

III. Collaborative Scheduling Software

Enabling widespread and crew-to-Earth collaboration on daily timelines as described in the preceding concept of operations requires major, but evolutionary, changes in scheduling software. In addition, a robust communication infrastructure designed for long round-trip communication delays is needed. Current development efforts are under way on delay-tolerant networking, internet-in-space, and publish/subscribe methods. The Appendix, at the end of this chapter, describes interplanetary networks and the message bus architectures. Other communication infrastructure elements are planned. Relay satellites will orbit Mars to provide communication for the 12 hours per Earth day during which the habitat will be on the far side. Additionally, a relay satellite in orbit around the sun will provide communication when the sun is directly between Earth and Mars (in worst cases, the sun-occultation direct communication path could be blocked for two weeks).

---

*On the ISS, task-list items are not submitted to the scheduling engine; the crew selects them and executes them whenever they want (the crew notifies the ground of their actions). For this reason, task-list items are limited to those that do not have difficult timing requirements, do not use scarce shared resources, and they are limited to tasks that are done by the crew. Additionally, the ISS crew cannot see the scheduling models of the tasks, only the procedures.

The salient features of a scheduling system that can maintain the principles of the collaboration-based concept of operations are use of standard terminology, a comprehensive modeling schema that represents all the constraints, an automatic scheduler that understands the models and produces a desired timeline, remote access to the scheduling system, a human interface that is user friendly to all users including the crew, and the ability to perform over a delay-tolerant network. Figure 3 shows the components of such a scheduling system and how they are used by the collaborators. There would be two instances of the scheduling system, one on Earth and another in space. The Earth-based instance supports widespread collaboration on the preliminary timeline—the ground support teams being the primary contributors and the crew offering limited contributions. The in-space instance supports collaboration on the final timeline with the crew being the primary contributors and the ground support offering limited contributions.

The crew will have little time to work on development of the timeline. Most equipment mode models and task models will be preconstructed before transferring them to the in-space software. Most of the timeline will be developed on Earth. After the locus of development moves to space, the ground can review timeline additions and changes made by the crew when made far enough in advance of execution, but the crew does not have time to “discuss” the changes via text messages. Some crew changes to the timeline will be in response to the ongoing situation and cannot wait for the light-time delay for review and oversight (these changes can be called real-time replanning). The modeling schema and the scheduling engine must synergistically and automatically produce valid timelines.

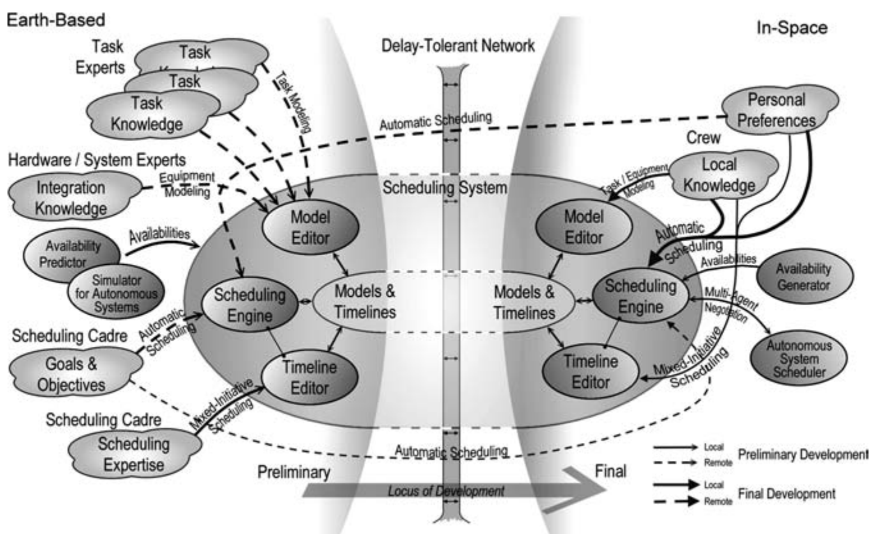


Fig. 3 Software and infrastructure.

A. Equipment Mode Modeling

In space, as on Earth, most tasks are accomplished using equipment. Most types of equipment have modes of operation, e.g., a microwave oven has defrost, reheat, and cook. The microwave oven's power requirement for each mode is predefined. On a space platform, the characteristics of each piece of equipment are well known to those building and integrating the equipment into the platform systems. The equipment and their modes may be modeled independently of the tasks that will use the equipment. Equipment mode models may use multiple resources and may use other equipment in specified modes. Equipment mode models can describe a hierarchy of constraints and alternate resources. Equipment mode models allow low-level resources such as power to be hidden from the task modeler. The task modeler can only request these low-level resources by selecting equipment that uses them. Using equipment modes to model resource usage is an extension of a method proposed by Hagopian and Maxwell [3].

Earth-based collaborators will build these models using a distributed-via-remote-access* feature of the scheduling system to build models in a central data repository. Using a record locking or record checkout method allows multiple users to work at the same time as long as they work on different equipment mode models; revision control is built into the system. All collaborators can view the data of other collaborators.

The in-space crew can view the models in the Earth-based repository via the delay-tolerant network. After the data are transferred to the in-space instance, the crew, using their knowledge of the local configurations, can make needed modifications to the models. The model viewing and editing software must be usable by the crew. The Earth-based support teams can review the crew's changes to the models by either remote display or by transferring a copy of the datasets back to Earth.

B. Task Modeling

A specification of the types of goals, states, activities, equipment (resources), and constraints necessary to achieve an objective is called a task model. Task models can range from simple to complex and can depend heavily on the capabilities of the scheduling system that will use them. For the most part, Earth-based task experts will construct, review, and test the models before flight. The modeling schema needs to represent all of the requirements (and their flexibilities) so that automatic scheduling can be employed; it also needs to have a straightforward representation so that the task experts (technicians, scientists, etc.) can easily enter the models; and it needs to be easy-to-use so that all collaborators, including the crew, can understand the models. A possible modeling schema has been previously proposed [4]. Some of the key features are the following:

1) Decomposition of the operations into salient components: Operations are decomposed into tasks that define resource requirements and sequences that define

---

*The function is distributed to remote collaborators by providing remote access to the central location. Accessing the central location could include automatic download and/or update of a local client. In-space software would not be automatically updated.

relationships between tasks. Sequences can also contain other sequences, repeated tasks and sequences, and optional tasks and sequences. Sequences can have multiple scenarios, that is, alternate ways to accomplish the same goal. Tasks can specify alternate resources and variable durations with preferences.

2) Rich expression of the relationships between components: Common-sense representations of temporal relationships use everyday concepts like follows, during, and overlap. Innovative enhancements to represent the continuance of resource usage between tasks, the interruption of tasks, minimal percent coverage, and temporal relationships to outside tasks have been added to the modeling schema. The timing of a relationship can have minimum, maximum, and preferred values.

3) Public services: The modeling schema also includes the concept of public services—models that are scheduled at the request of another model.

The collaborators use the same type distributed-via-remote-access features as the equipment mode model builders to access a central data repository. Normally, all collaborators can view the data of other collaborators; however, models containing sensitive information may be hidden to some collaborators, but never to the crew. The builders of task models can submit their models to the scheduling engine to test the models for validity and usability. Depending on the concept of operations, the results of scheduling can be part of the developing timeline or they can be tested only.

Like equipment mode models, task models can also be viewed by the in-space crew using the delay-tolerant network. After the data are transferred to the in-space instance, the crew members, based on their local knowledge or personal preferences, can modify the task models. Complex syntactical languages, difficult-to-navigate user interfaces, or a modeling schema that does not match the real world could render such a system nearly useless to the crew. The Earth-based support teams can review the crew's changes to the models by either remote display or by transferring a copy of the datasets back to Earth.

C. Automatic Scheduling

Automatic scheduling will be the primary mode of developing the task timelines. An incremental scheduling engine has characteristics that enable supporting multiple collaborators simultaneously contributing to the development of one timeline. An incremental engine is fed by a queue of scheduling requests (sequences of tasks). The engine adds each scheduling request to the timeline without adjusting the times of already scheduled tasks and without introducing constraint violations or resource overbooking. An incremental engine implements both of the enabling principles of the concept of operations—it adds to the timeline without invalidating what is already on the timeline and it does not require a user to have global expertise on the items in the timeline. The core logic of an incremental engine is usually an implementation of a greedy algorithm [5]; that is, it makes choices based only on scheduling the current request. These engines may use analytical, heuristic, algorithmic, and/or artificial intelligence techniques. Incremental scheduling engine usage is depicted in Fig. 4. Some scheduling requests are emphasized to show that they may be very complex. The figure also shows that multiple queues from multiple remote users may be merged into a single queue.

As an incremental scheduling engine processes a sequence, it will search for a near-optimum placement for the multiple tasks of the sequence being scheduled.

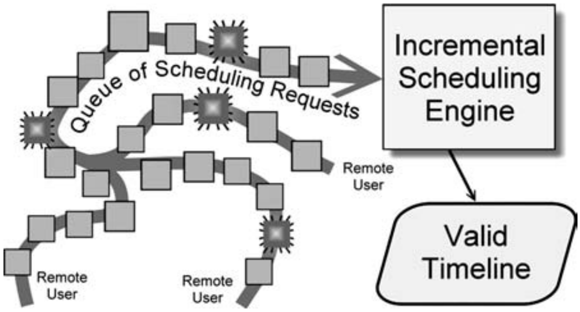


Fig. 4 Incremental engine usage.

(Incremental engines do not provide global schedule optimization.) The tasks always have temporal relationships to each other and may share the same resources. However, all of the tasks scheduled by previous scheduling requests are locked, and the residual resource profiles are treated as initial resource profiles for the current request.

Incremental engines can process a request to delete the results of a previous scheduling request. A request to delete an item on which another item depends (reverse dependency) will fail. Deleting only the requested items would create an invalid timeline; deleting dependent items would violate the tenant that processing a request does not affect what is already on the timeline. Of course, after reviewing the failure report, the user could formulate a deletion request to include all of the dependent items. Deletion requests are moved to the head of the queue to free up resources and reduce the likelihood of a dependency being established.

Incremental engines can also process a request to replace a previous scheduling request. When replacing items on the timeline, resources are reused and reverse dependencies become additional constraints. For example, suppose sequence A and D were scheduled by different requests. Suppose sequence A contains task T and sequence D had a requirement to be scheduled after task T. Further suppose that a request is made to replace sequence A with sequence B. Then B must include a replacement task T and the engine will schedule sequence B so that the replacement task T is scheduled before sequence D.

Collaborators use remote access to submit scheduling, deletion, and replacement requests from the repository of models. Multiple collaborators can submit to the queue simultaneously. The scheduling system provides the ability for all users to see the current timeline as it is being developed. The crew can used the delay-tolerant network to submit scheduling requests and to view the timeline.

Incremental scheduling engines and their usage are discussed in detail by Jaap and Phillips [6].

D. Mixed-Initiative Scheduling

Mixed-initiative scheduling refers to building a timeline using a timeline editor, i.e., it is a manual process. Mixed initiative is used when the contributor knows requirements that are not described in the scheduling requests, the scheduling

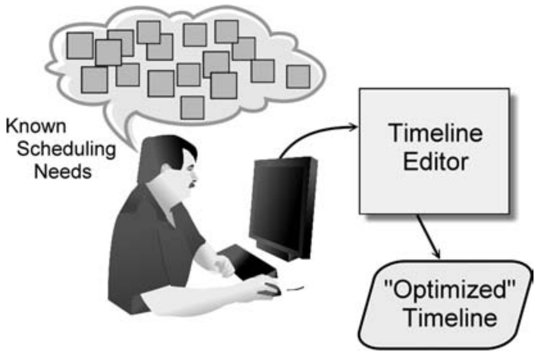


Fig. 5 Mixed-initiative scheduling.

engine is weak, only a few new requests are to be added to the timeline, or the user wants to fully control the results (Fig. 5). Reckless usage of mixed initiative scheduling will violate the enabling principles of the concept of operations—moving already scheduled items can invalidate what is already scheduled or require the user to have global knowledge of the scheduling requirements. (Mixed-initiative scheduling does not automatically provide global schedule optimization. However, if the user is an expert and the problem is straightforward, global optimization might be achieved.) Mixed-initiative scheduling is also discussed by Jaap and Phillips [6].

Mixed-initiative schedulers usually have logic to help the user avoid violating constraints and allow the user to override constraint limits. If the models are complete, the editor might invoke iterative-repair logic to move other tasks and eliminate constraint violation introduced by a manual edit. The editor might invoke an incremental scheduler to make slight adjustments to the user's input; this feature is called *snap-to*. Additionally, the editor might use an incremental engine to suggest times where tasks could be placed without introducing constraint violations.

The timeline editor will display considerable information about the timeline, the models, and the availabilities. It will be closely coupled to the data repository and the scheduling engine. It will not operate over the delay-tolerant network and will be available only to local users. During the development of the preliminary timeline, a cadre of highly skilled users, called the scheduling cadre, will be the primary users. After the locus of development moves to space, the crew can use mixed-initiative scheduling, but will do so rarely.

E. Resources, Conditions, and Autonomous Systems

Scheduling depends on the availabilities of resources that are scarce and shared, on conditions that occur intermittently, and on the available of other, possibly autonomous, systems. Power, cameras, and storage space are examples of resources; daylight, communication links, and sand storms are examples of conditions; and self-scheduling robots and rovers are examples of autonomous systems. During the development of the preliminary timeline, availabilities are

predicted by Earth-based software, and autonomous systems are simulated. During the finalization of the timeline in space, the availabilities of resources and future conditions are predicted by adjunct software. For example, on Mars a weather prediction program will warn of approaching sand storms.

The in-space scheduling systems will negotiate with the rovers and robots to jointly arrive at a common timeline. The negotiation might go as follows:

1) The main scheduling system will ask a robot if it is available to do a certain task at a certain time.

2) The robot scheduler will attempt to the schedule the requested task; if possible, give an affirmative answer; if not, an alternate time might be suggested.

3) After the main scheduler has found acceptable times for all tasks of the scheduling request, it would send the robot a commit message, and the robot would complete its scheduling.

F. Terminology and Standards

Historically, the planning and scheduling for space domain has been fragmented along human vs non-human flights and also by the sponsor of the missions (NASA center, ESA, etc.). The fragments have each evolved their own terminology and standards for expressing planning and scheduling requirements, results and displays. Many of the differences are semantic—a consumable resource and a depletable resource have the same characteristics, a state is a condition, etc. Some of the differences are substantive—activities have durations and are delimited by start and stop events. In one scheme, temporal relationships may be expressed relative to the activity; in another scheme, they are relative to the delimiting events. “Activity A occurs during activity B” can be directly expressed in one scheme but in the other becomes “start of B occurs on or after start of A and stop of B occurs before or on stop of A.” Goals may be chains of events or sequences of activities. Similar differences exist for the results and the displays. Human missions to the moon and Mars will bring together contributors from many backgrounds because the missions will include humans, robots, rovers, automatic systems, and remote-commanded systems.

Collaboration by a large diverse group of ground support persons and by the crew will require a standard terminology be adopted by all. It is especially important that the crew, at a minimum, be able to look at any timeline (for any system), understand it, comment on it, and, in some cases, modify it. The crew is dedicated to planning and scheduling and should not devote time and effort to learning different scheduling terminologies or methods. Fortunately, existing standards bodies, such as the Consultative Committee for Space Data Systems (CCSDS), are available to apply their expertise in establishing, negotiating, and documenting the needed data standards.

G. User Interfaces

Good user interfaces with the planning and scheduling software are necessary for successful collaboration between diverse contributors. The user interface needs to allow those who are not experts in planning and scheduling to perform as “virtual” experts. Like the driver of a car who is able to steer the car without

concern for how to manipulate the steering wheel, the user of the planning and scheduling software should be able to produce the desired timeline without concern about the user interface.

To support widespread collaboration and remote access across the solar system, the scheduling software user interfaces will need some special features. One might say the software must have delay-tolerant user interfaces. Delays will be caused by light-time and by loss of signal to relay satellites. Some of the ameliorating features of the software are 1) maximizing local processing to avoid delays, 2) continuing to interact while waiting for a reply to a message, and 3) providing visibility into the stack of sent messages showing time remaining until a response is expected.

The crew will have special user interface needs. The console in the habitat or vehicle may have the standard user interfaces, but the interfaces used when outside will have special requirements. Figure 6 shows a PDA-like device attached to a crew member's sleeve. This device is in communication with the planning and scheduling system in the habitat. The crew member should be able to look at the timeline and possibly modify the timeline using the sleeve-attached device.

IV. Conclusion

As humans explore regions of space where the Earth appears as a mere point of light, and round-trip communication delays exceed half an hour, it will be imperative that the humans on the journey exercise significant control over their daily timeline and the timelines of their companion systems. Yet the crew does not have the time to do all of the scheduling. Scheduling will be a collaborative effort between multiple ground support teams and the crew.

A proposed new concept of operations is based on the principles that adding to the timeline does not invalidate what is already on the timeline and contributors do not need global expertise about the items on the timeline. The concept calls for developing the preliminary timeline on Earth, having the Earth-based teams as the primary contributors and the crew providing only minimal input. Once the preliminary timeline is uplinked to the in-space location, the crew will become the primary contributors, and the Earth-based teams will provide only minimal input.

Fig. 6 User interface for the crew.

This concept is tailored to function effectively in the delayed communication environment.

Planning and scheduling software will evolve to meet the needs of the new concept of operations. Automatic scheduling will become the normal way to produce the timeline. Modeling will be partitioned into equipment mode modeling and task modeling so that different experts with different knowledge can contribute effectively. An equipment mode model will reflect the different modes in which the equipment operates and will book all resource and condition requirements. Task models will be grouped into sequences showing the temporal relationships of the tasks. An incremental approach to scheduling will enable multiple experts to schedule sequences without impacting sequences scheduled by other experts (the crew being among these experts). Because the models reflect all of the requirements and the scheduling engine can produce good timelines, mixed-initiative scheduling will be used rarely.

The synergy of the new concept of operations, delay-tolerant communication and software (including user interfaces), and advances in scheduling technology will allow the in-space crew and Earth-based support teams to collaborate on the scheduling of the tasks done by the crew, the operation of the vehicle or habitat, and the operation of the various near-autonomous robots and rovers.

Appendix: Delay-Tolerant Networking

Planning and scheduling is only one application that will depend heavily on a communications framework that will carry data between Earth and Mars or between. Without such a framework, in-space collaborative scheduling would be impossible. The cost of spaceflight being prohibitive as it is, much work will be done to leverage existing technologies and commercial hardware and software to use them and manipulate them to work in these foreign environments. Accomplishing this will necessitate standardizing communication between applications using concepts like a message bus, which travels over ubiquitous protocols such as the internet modified for light-time delay.

A. Interplanetary Internet

Standardized digital communication devices have revolutionized the way we communicate and interact with information, but TCP/IP, as implemented on Earth, does not translate well in a space environment where line-of-sight connectivity is intermittent, communications have significant delays, and data may have high error rates requiring retransmissions. However, a TCP/IP-based network can be established at Mars, and, due to the underlying TCP/IP technology (breaking messages into independently transmitted packets, retransmission of only bad packets, and reassembly of the final message at the destination), the Earth-based network and the Mars-based network can be interconnected. Ongoing research is spawning a new concept, delay-tolerant network (DTN), to mitigate the shortcomings of TCP/IP. DTN will probably employ concepts such as store-and-forward message delivery.

In store-and-forward, relay systems capture incoming data and store them locally before transmitting them to the next link in a chain (Fig. 7). If there is a transfer failure, the data will not have to travel the complete distance again. DTNs can be

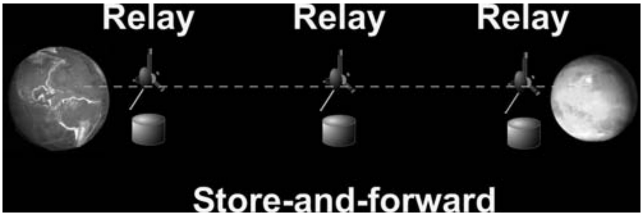


Fig. 7 Inter-planetary Internet.

used to route data between two traditional TCP/IP networks almost transparently to the TCP/IP packets concept. However, many of today's applications are not designed with such communication delays in mind. Loss of continual heartbeats or continuous data streams would cause many of today's applications to fail as they wait for the next packet to arrive. Therefore, any applications themselves must be built with data delays in mind. The applications must be more robust and fault tolerant and should not malfunction when there is a long data delay.

B. Message Bus

Message buses standardize information exchange as well as data pathways between two end points. Buses can be layered upon DTNs to provide more abstraction between the software and the communication infrastructure, allowing for the flexibility of architecture upgrades. Most message bus modes include a publish/subscribe protocol, sometimes called pub/sub, in addition to more direct request/reply connections.

In the publish/subscribe mode, client applications subscribe to information they would like to be provided. As messages pass by, messages that match the requested type are captured by the client API and passed to the application.

References

[1] Compton, W. D., and Benson, C. D., "Living and Working in Space: A History of Skylab," NASA Headquarters, Washington, DC, June 1983.

[2] Jaap, J., and Muery, K., "Putting ROSE to Work: A Proposed Application of a Request-Oriented Scheduling Engine for Space Station Operations," *Proc. Sixth International Conference on Space Operations (SpaceOps 2000)*, Toulouse, France, June 2000.

[3] Hagopian, J., and Maxwell, T., "Explicit and Implicit Resources: A Simplified Approach to User Requirements Modeling," *Proceedings of the Fourth International Symposium on Space Mission Operations and Ground Data Systems*, ESA SP-394, European Space Agency, Paris, Sept. 1996.

[4] Jaap, J., Davis, E., and Richardson, L., "Maximally Expressive Modeling," *Proceedings of the Fourth International Workshop on Planning and Scheduling for Space*, European Space Agency, Darmstadt, Germany, June 2004.

[5] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, 2nd ed., Sec. 16, MIT Press, Cambridge, MA, 2001.

[6] Jaap, J., and Phillips, S., "On Using an Incremental Scheduler for Human Exploration Task Scheduling," *Proceedings of the 2005 IEEE Aerospace Conference*, Big Sky, MT, March 2005.

Chapter 22

Some Results on Authentication and Encryption Schemes for Telecommand and Telemetry Data

S. Spinsante,* F. Chiaraluce,† and E. Gambi*
Universitá Politecnica delle Marche, Ancona, Italy

Nomenclature

C_i = ith block of cipher text
$g(x)$ = error correcting code generator polynomial
I = interleaver depth
$i(.)$ = identity function
K = shared secret key in a cryptographic algorithm
n = dimension of the data block given as input to a block cipher, bits
P = probability of correct received code word
p = channel error probability in the case of sparse errors
p_B = channel error probability in the case of burst errors
P_i = ith block of plain text
$s(x)$ = error correcting code syndrome polynomial
$r(x)$ = received code word polynomial

I. Introduction

RECENTLY the Consultative Committee for Space Data Systems (CCSDS) Security Working Group (WG) has been actively engaged in determining security recommendations for space missions, thus filling a gap in the CCSDS activities. Security of data communication in space systems has not received, up to now, enough attention, most of all in the case of civil missions. To date, missions falling into this category have basically relied on uniqueness and unavailability of publicly

---

*Doctor of Engineering in Telecommunications.
†Assistant Professor of Telecommunications.

known technical details to deter unauthorized accesses. On the contrary, military space missions have traditionally included a high level of built-in security.

This situation is currently changing, because of the increasing number of international missions that require cross-agency support, and to the real possibility of using public ground data networks to transfer mission control and monitoring data.

The expansion of network interconnectivity for data dissemination and mission scheduling creates new and additional threats against civil space missions; they should be analyzed to provide suited countermeasures for the protection of assets and critical services. Among the threats, we can cite unauthorized access, which is a potential threat against all mission types, and data corruption. Threats to spacecraft telecommand and telemetry links originate in their transmission through the physical radio frequency (RF) medium. These transmissions are potentially subject to detection and interception by entities unauthorized to receive the information. As a result, there is a possibility that the information carried in the transmissions is improperly exploited. A particularly dangerous active threat might allow an unauthorized entity to transmit telecommands that might cause accidental or malicious damage of the spacecraft, or even its destruction. With the large investment needed to deploy space missions, any process that may counter the threats to the spacecraft is desirable and will be essential for some types of space mission. A similarly dangerous active threat would be the jamming of the RF medium by an unauthorized entity, to block entirely transmissions to or from the space link. This way, access is denied, which could result in loss of science data, telemetry, or telecommand, and eventually causing unrepairable damage to a spacecraft. Threats to the space mission ground data system can be considered the same as the information security threats to any open or private computer network.

An overview of the security services that could be required in the framework of a space mission is provided in the revised version of the CCSDS Green Book on systems security [1]. To define sets of generic security requirements for different space mission types, missions have been classified as requiring high, moderate, or minimal levels of security. This is because different mission types will require the implementation of different security services to satisfy specific mission requirements. In addition, different mission types may require different assurance of correctness of operation for each security service. Basically, the two main security strategies involve authentication and encryption functions, which are applied to telecommand (TC) and telemetry (TM) data, respectively, being authentication and integrity the two security services required in almost all of the mission classifications. Authentication and encryption algorithms should be selected according to the unique properties of the space environment. Thus, for example, highly asymmetric communications and significant round-trip delays should lead to the adoption of low time-consuming algorithms with reduced overhead. Further, high error rates frequently experienced on a space link suggest the selection of encryption and authentication schemes that are also robust to error propagation.

Though based on such relevant premises, however, the topic has been often faced in the past in merely qualitative terms, and only a very few studies have been carried on with the aim to provide a quantitative performance evaluation of available security solutions, suited for space applications. Explicitly, the authentication and encryption algorithm trade studies promoted by the CCSDS Security

WG [2, 3] provide useful guidelines about the state of the art in authentication and encryption schemes, but to the authors' best knowledge no numerical results are yet available for an objective comparison of these schemes when applied to space data and their own formatting requirements.

Along with these considerations, this chapter presents some numerical results related to the evaluation of authentication and encryption schemes when applied to space data, in conjunction with error correction strategies that are already defined by the standards and currently used. Objective results, to compare some of the solutions now under investigation by the CCSDS, both in the case of TC authentication and TM encryption, are provided, by extending some preliminary evaluations [4] on the performance of Advanced Encryption Standard (AES)–based schemes when applied to TM and TC data structures. An important aspect concerns the impact on the encryption and authentication processes of residual errors introduced by the channel and not compensated by the coding layer of the TC or TM architectures. Therefore, there is the need to investigate the interactions between the encryption services and the forward error correcting (FEC) services that coexist in the system.

The results presented in this chapter have been obtained by considering randomly generated TC and TM frames. Even though this choice can affect the output of the applied authentication or encryption algorithm, it should not have consequences on the evaluation of the channel effects, which are mainly related to the way in which errors are spread over the transmitted binary sequences. To provide evaluations as reliable as possible, several simulations have been executed for each error probability under test, so that the final outcomes can be viewed as global indicators of the average system behavior.

II. TC Authentication

Telecommand and telemetry signals are an essential part of any space mission. TCs flow, together with ranging tones (the latter to determine the distance of the spacecraft or satellite with respect to a reference point), from the ground to the spacecraft; TMs flow, together with ranging tones and (often) payload data, from the spacecraft to the ground. A space mission might be jeopardized because of illicit TCs inserted in unprotected links. Also, accidental disturbances, like noise or human errors, can cause malfunctions of the system, while another relevant contribution derives from jamming. External attacks can aim to violate confidentiality in the transmitted data (passive traffic analysis) and/or to prevent providing a service (like with RF interferences) and/or to insert illegal TCs, to modify or reply intercepted legal TCs (impersonation attacks). Actually, attacks of the third kind are often the most dangerous ones, and require special care in the TC authentication. The problem seems to become more and more important since current links are prone to use extensively ground stations interfaced with open networks (i.e., Internet) intrinsically characterized by high vulnerability.

Nearly 15 years ago, the ESA had standardized an approach to telecommand authentication within its Packet Telecommand Standard [5]. Though in the past this approach was adopted and implemented within space-qualified chip sets integrated into ESA spacecrafts, now it can be no longer considered as a standard. Modern, high-speed processors and flaws in its foundation technology (Knapsack) have relegated this authentication procedure as historical.

Authentication solutions based on the well-known mechanism of message authentication codes (MACs) have been extensively revisited in recent studies. While digital signature schemes use public/private key pairs, MACs use a shared secret key K to provide both authentication and integrity in several different ways. As an example, a check word can be created over the data with an embedded secret key, or the check word is created by a hash algorithm and then encrypted using the secret key. The classical MAC construction is based on the combination of hashing and encryption [typically cipher block chaining (CBC) and cipher feedback chaining (CFB)]: a check word is created over the data using the hash algorithm, then by means of the encryption algorithm and the secret key, the check word is made secret. At the receiver, where the secret key shall be known, the check word is regenerated on the received data and, at the same time, the received encrypted check word is decrypted, in such a way as to compare the regenerated and received check words in clear. If and only if they are identical, the receiver is assured of the authenticity and integrity of the received data.

In this section we discuss the TC authentication issue, by testing the performance obtained when the Advanced Encryption Standard (AES) is adopted in the two different operational modes previously mentioned, i.e., CBC and CFB. AES has recently replaced the Data Encryption Standard (DES) in many conventional applications. A first remark coming from CCSDS concerns the observation that while DES is usually very weak as an encryption algorithm, because of the processing power available to modern computers, a DES-based MAC algorithm might have satisfactory performance under some circumstances. However, with the generally agreed deprecation of DES, it will not be in wide use anymore and will not be a convenient algorithm to adopt for a MAC. On the contrary, the efficiency and security of AES is well known and does not need additional demonstration or verification. Thus, we aim to test the behavior of the authentication process in the presence of transmission errors due to the channel, a relatively new topic; more specifically, we are interested in simulating the effect of sparse channel errors.

A. CBC MAC Authentication

The most common MAC architecture involving the use of a block cipher implements the so-called CBC mode. In CBC-MAC authentication, the message signature is the output of a block cipher applied in CBC operational mode on the message, as shown in Fig. 1, for a given value of the initialization vector (IV). Changing the IV will change the final output of the scheme.

The main target of the authentication process is to prevent the use of illicit and disruptive TCs by the receiving unit: TCs carrying a signature that cannot be verified at the receiver can be simply discarded. However, authentication does not provide solutions against the so-called replay attacks; they can be avoided by inserting a suited counter (TC counter), which is incremented each time a new TC is transmitted. As a matter of fact, the structure of the TC segment currently adopted by ESA includes the so-called logical authentication channel (LAC) counter field that has been maintained within the data structure adopted throughout our simulations, as reported in Fig. 2.

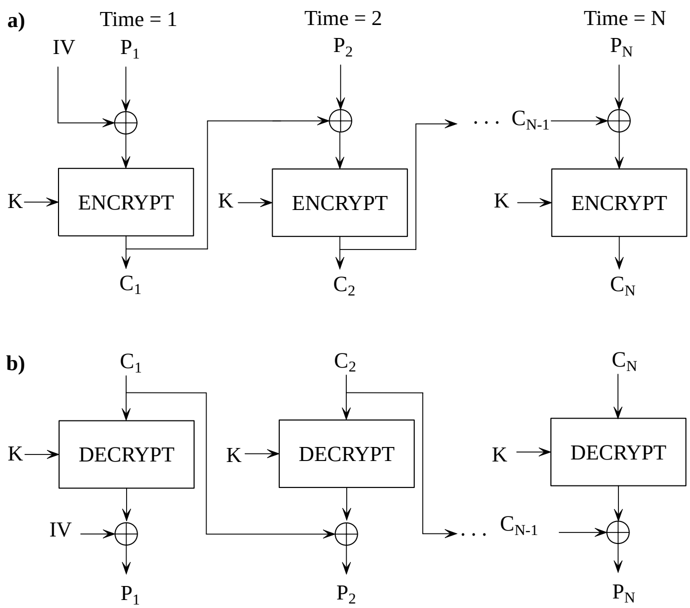


Fig. 1 Cipher block chaining mode: a) encryption, b) decryption.

In the evaluated authentication solution, the block cipher adopted is AES, with a block dimension of 128 bits. At each stage, a block of 128 bits of plain text P_i (i.e., a portion of the TC segment) is exclusive-OR-ed (XOR-ed) with the result of the previous ciphering stage; in the first stage, the first block of plain text is XOR-ed with the IV. The following steps describe the overall procedure applied to a TC segment:

1) The message to cipher is split into blocks of n = 128 bits. If padding is needed to have a total length multiple of 128 bits, a single 1 followed by a suited number of 0s must be inserted at the end of the TC segment.

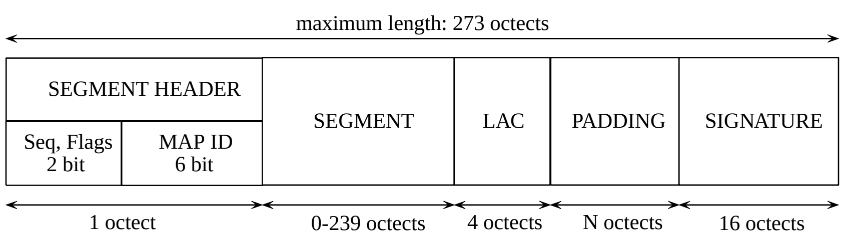


Fig. 2 Structure of the TC segment used for simulation purposes (MAP: multiplexed access point).

2) The IV value is obtained by a pseudo noise (PN) generator. IV is XOR-ed with P_1, the first block of 128 bits in the TC segment. IV and K should be different to provide increased security.

3) The output of each ciphering stage is a block of 128 ciphered bits, C_i; it is passed as input to the next ciphering stage, repeating this step for each input block P_i.

4) Calling C_N the output block of the last stage, we have that MAC (TC segment) = $i(C_N)$. In general, $i(.)$ is the identity function, but it is possible to define a different output transformation.

B. CFB MAC Authentication

The CFB operational mode is slightly different from the CBC one. As with CBC, the units of plain text are chained together, so that the cipher text of any plain text unit is a function of all of the preceding plain texts. The main difference with respect to CBC is that in the CFB mode, at each stage, the preceding cipher text is used as input to the encryption algorithm to produce a pseudorandom output, which is XOR-ed with the plain text unit, to produce the next unit of cipher text. Typical applications of the CFB operational mode include general purpose stream-oriented transmissions and authentication. In a generic J-bit CFB mode, the input to the encryption function is a shift register initially set to some initialization vector IV. The leftmost J bits of the output of the encryption function are XOR-ed with the first unit of plain text P_1 to produce the first unit of cipher text C_1, which is then transmitted. In addition, the content of the shift register is shifted left by J bits and C_1 is placed in the rightmost J bits of the register. This process continues until all of the plain text units have been encrypted. For decryption, the same scheme is used, except that the received cipher text unit is XOR-ed with the output of the encryption function to produce the plain text unit. The overall CFB mode is depicted in Fig. 3.

C. Forward Error Correction Scheme for Telecommand Data

The reliable error-controlled data channel, through which TC data may be transferred, is provided according with the CCSDS specification [6]. Data are encoded to reduce the effects of noise in the physical layer; a modified Bose-Chaudhuri-Hocquenghem (BCH) code has been chosen for this protection. The codeblock is produced using a systematic coding technique that generates 7 parity check bits computed from 56 information bits. A filler bit (zero) is appended at the end. More precisely, the code used is a (63, 56) expurgated Hamming code, whose generator polynomial is given by

$$g(x) = x^7 + x^6 + x^2 + 1 \tag{1}$$

This code has a minimum distance equal to 4, which implies it is either a triple-error detecting or a single-error correcting and double-error detecting (not both). The encoder implementation is shown in Fig. 4: the shift register is initialized with zero, the switch is in position (1) while the 56 information bits are transmitted, in position (2) for the 7 parity bits and in position (3) for the appended filler bit.

For simulation purposes, we refer directly to the channel error probability p, instead of the conventional signal-to-noise ratio. Decoding is based on a hard

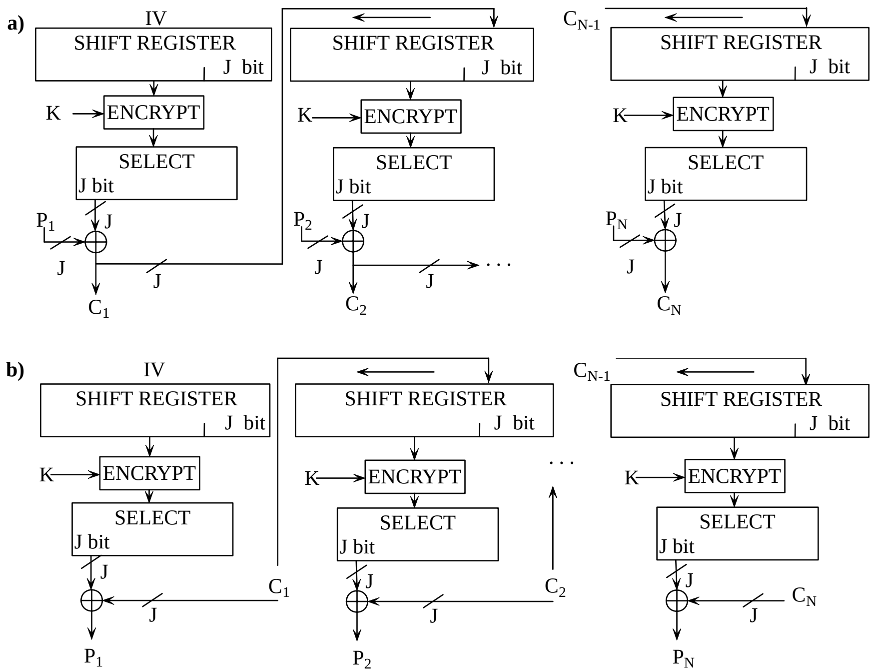


Fig. 3 Cipher feedback mode: a) encryption, b) decryption.

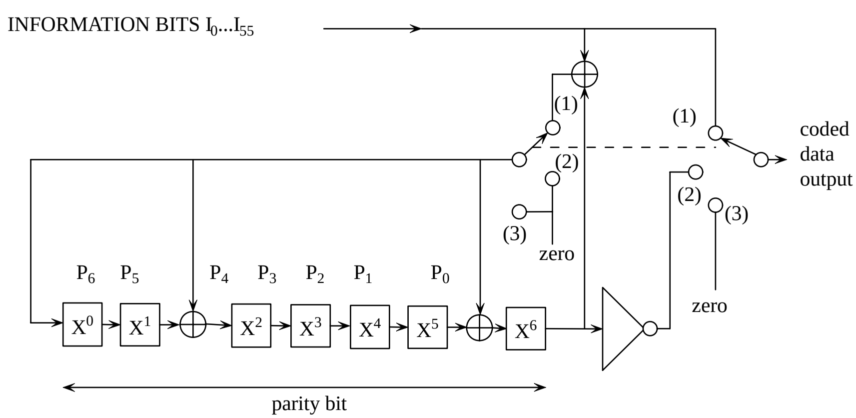


Fig. 4 Encoder circuit.

decision, and exploits the calculation of the syndrome polynomial $s(x)$. Noting by $r(x)$ the polynomial representing the received code word after the hard-limiter, $s(x)$ is computed as $s(x) = \text{mod}\,[r(x)/g(x)]$.

Sixty-three different syndromes correspond to as many single-error events. We assume to adopt a bounded-distance decoder, which means that the decoder declares a decoder failure any time it finds a syndrome different from the 63 corresponding to correctable errors. Obviously, this does not prevent having decoding errors that occur when errors transform the transmitted code word in another code word or a sequence at unit distance from another code word. In the case of an additive white Gaussian noise (AWGN) channel, characterized by sparse random errors, the probability that the received code word is correct (i.e., it does not include more than 1 error) can be computed as

$$P = (1 - p)^{62} \cdot (1 + 62 \cdot p) \tag{2}$$

with p being the probability of a single error. On the contrary, in the case of burst errors, no simple formula is available, and simulation is necessary to estimate performance.

D. TC Authentication: Simulations and Results

To find the correct authentication rate (CAR), i.e., the number of authenticated TC segments that are verified at the receiving side, which is possible to obtain when applying AES-based MAC authentication on TC data, in the presence of sparse errors on the channel, we have implemented a software simulator, whose architecture can be summarized in the following list of functionalities:

1) Start by generating an arbitrary TC segment, having a length variable from 2 to 239 bytes.
2) Generate IV and K.
3) Append LAC value and padding, if necessary.
4) Generate MAC.
5) Simulate the channel effect, through the assignment of a value of p.
6) Compute MAC on the received TC segment.
7) Perform MAC verification.
8) Save the result on file and increase the LAC counter value.
9) Repeat the process for a new TC segment, or exit.

At first, we have tested the performance obtainable by applying the two AES-based MAC schemes to authenticate TCs of different lengths (2, 4, 8, 32, 128, 184, and 239 bytes). The TCs have been chosen at random within the output of a PN generator. For each TC length, the long simulation needed to ensure a desired statistical confidence level for the results has been realized by repeating continuously the same TC. Intuitively, this is the most favorable situation for an opponent that can work having available a large number of plain text/MAC couples. Figures 5 and 6 report the CAR (in percent values) that are obtained by adopting the AES-CFB MAC and AES-CBC MAC authentication schemes, respectively, when no channel coding is applied, in the case of sparse errors, for p ranging between 10^{-8} and 10^{-4}. TC-x denotes telecommand of length x.

As evident in the figure, when the channel error probability gets around 10^{-6} or lower, practically all of the authenticated TC segments are verified at the receiver,

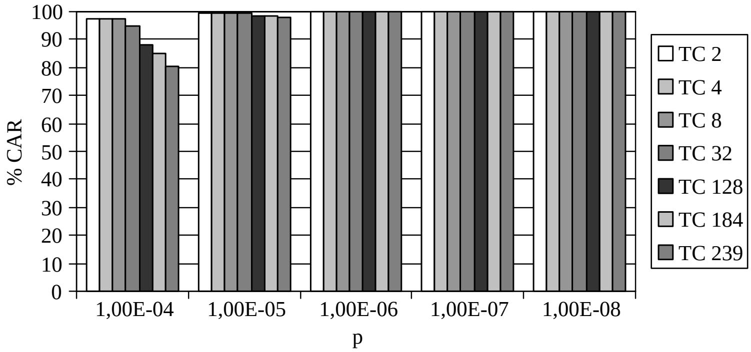


Fig. 5 Percent CAR values in the case of AES-CFB MAC authentication, no modified Hamming code applied.

thus providing a CAR value of 100%. In this case, the very low number of errors on the channel does not influence the authentication process. On the other hand, for higher values of the channel error probability, the CAR goes under 100%, even if it maintains quite high performance, which is slightly better in the case of AES-CFB MAC than in AES-CBC MAC. This can be explained by considering that the CFB mode limits the error propagation on the received TC segments, during recalculation of the MAC, which is then compared to the received MAC value, to verify the authenticity of the TC frame.

Similar simulations have been performed to evaluate the CAR values at the receiver, when the modified Hamming code is applied. The results obtained are shown in Figs. 7 and 8 (note the different scale with respect to the previous figures).

As reasonable and expected, the percent CAR values get higher and higher, approaching 100% even for (relatively) large channel error probability. This happens because the adoption of the modified Hamming coding scheme allows the

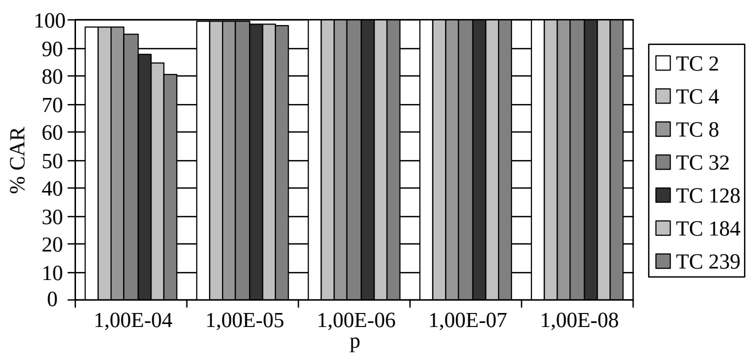


Fig. 6 Percent CAR values in the case of AES-CBC MAC authentication, no modified Hamming code applied.

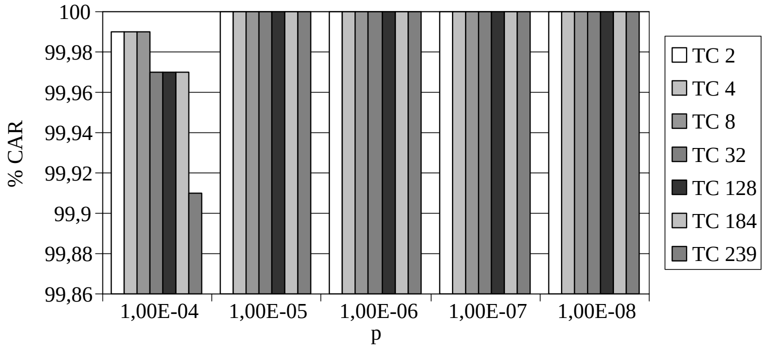


Fig. 7 Percent CAR values in the case of AES-CFB MAC authentication, with modified Hamming code applied.

removal of almost all the channel errors before the signature verification process at the receiver takes place. There are no substantial differences between the use of CFB and CBC MAC schemes in this case; then, the choice between them should be eventually guided by other constraints on the authentication system design.

TC encryption, as a possible alternative to TC authentication, has been rarely proposed, up to now, for this kind of application. On the other hand, especially in the case of short TCs, like those used for routine spacecraft maneuvering, an encryption key could be easily extracted from a sequence of encrypted and plain text pairs. Maybe, military missions could be the only ones really concerned about total data obscurity at the transfer level. A very few examples of possible TC encryption solutions are provided in the available literature, mostly suggesting the adoption of DES as encrypting function. Further analysis is therefore necessary, either on the adoption of more robust encryption schemes, like AES or elliptic

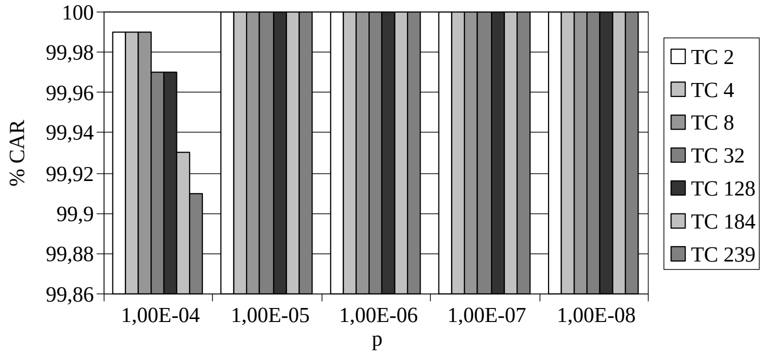


Fig. 8 Percent CAR values in the case of AES-CBC MAC authentication, with modified Hamming code applied.

curve cryptography (ECC), or on the evaluation of related issues, like key sharing and distribution, and suitable location of the encrypting functions within the CCSDS protocols architecture.

III. TM Encryption

TM encryption should be adopted in all missions requiring high security for satellite telemetry (i.e., navigation and communications). Because of the huge amount of TM data to be protected and transmitted during a mission, symmetric encryption schemes should be adopted, as computationally more efficient than asymmetric schemes. As previously discussed, AES is preferred to DES for its robustness and greater efficiency; with various operation modes, it can work as a stream cipher. In a TM stream, we typically have a series of TM transfer frames separated by the synchronization marker field and the channel coding field. In the currently suggested encryption procedure of TM frames, the whole frame, comprising the frame header (6 octets), the transfer frame data field (whose length is mission specific), the operational control field (4 octets), and the frame error control field (2 octets) are encrypted at the security sublayer of the data link layer, while the synchronization marker field and the channel coding field added at the coding sublayer, before transferring the data structure to the physical layer, are not encrypted.

The encryption of a stream of TM frames can be implemented by using either a self-synchronizing stream cipher mode, such as cyclic text auto key (CTAK), also known as CFB, or a key auto key (KAK) mode, also known as output feedback (OFB). CFB operational mode, which has already been presented in Sec. II, is a self synchronizing mode that does not require block padding, but it suffers from error propagation; self synchronization is obtained at the cost of incorrect decryption of the first frame, which is consequently discarded by the device processing the telemetry data. Moreover, denoting by J the number of bits that are shifted inside the register at each stage, and by n the length of each input block, to correctly decipher a block, it is necessary that all the previously received n/J blocks are correct. Thus, error propagation during the decryption phase involves n/J blocks after the block affected by one or more errors; for this reason we can define an error propagation rate as $n/J + 1$.

A. OFB Encryption Algorithm

The alternative scheme proposed for TM data processing is OFB, which is similar to CFB, except that the input to the encryption algorithm is the previous AES output. This way, the AES encryption engine at stage i produces an intermediate value O_i, which is XOR-ed with the input plain text unit P_i to output the corresponding cipher text unit C_i, and, at the same time, is used to feed the shift register of the following stage. In the decryption process, the intermediate value is replaced at each stage by the corresponding cipher text unit. The OFB state encryption function can operate independently of the plain text, thanks to an internal feedback mechanism that ensures no error propagation. In practice, a single bit error in the cipher stream can lead to a single bit error in the deciphered data; this is also the reason why this mode is often applied in stream-oriented transmissions over noisy

channels, like satellite communications. However, this scheme requires some means of providing synchronization between the encryption and decryption processes. The disadvantage of OFB is that it is more vulnerable than CFB to a message stream modification attack: complementing a bit in the cipher text reverses the corresponding bit in the recovered plain text. Thus, controlled changes to the recovered plain text can be made. The overall scheme of the OFB operational mode is shown in Fig. 9.

B. Forward Error Correction Scheme for Telemetry Data

We address the encryption solutions previously described, which are currently under evaluation by the CCSDS, and test their performance in the case of sparse

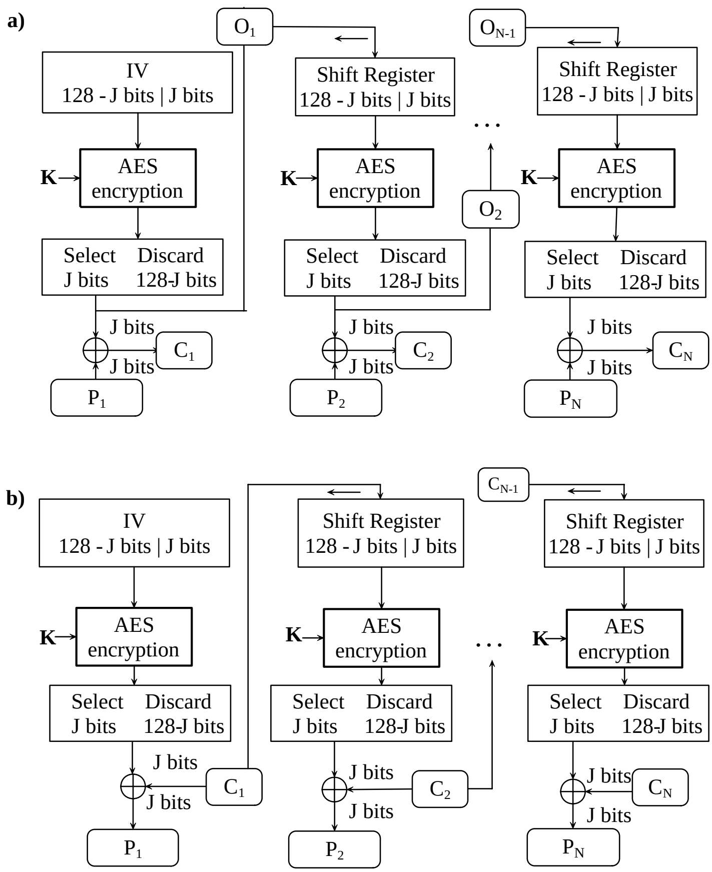


Fig. 9 Output feedback mode: a) encryption, b) decryption.

and burst errors. Sparse errors are simulated for different error probabilities, considering an AWGN channel, as in the case of TC data.

In the case of TM data, CCSDS specifies [7] three types of error control codes: convolutional, Reed Solomon (R-S), and turbo codes. For physical channels that are bandwidth constrained and cannot tolerate the increase in bandwidth required by the convolutional codes, the Reed Solomon codes have the advantage of smaller bandwidth expansion and the capability to recognize the presence of uncorrectable errors. The Reed Solomon code described within the CCSDS recommendation is a powerful burst error correcting code, with an extremely low undetected error rate. One of two different error correcting options may be chosen, which can correct 16 R-S symbols or 8 R-S symbols per code word. The two options cannot be mixed in a single physical channel. The selected R-S code is a systematic code, with a maximum code block length of $255 \cdot I$ symbols, where I is the interleaver depth, which can assume different values and is normally fixed for a given physical channel. Further details on the R-S coding scheme can be found in [7].

The TM frames transferred along a given physical channel must have a fixed length during a mission phase. This value must be compliant with the adoption of the R-S coding. CCSDS fixes, for cross support purposes, a length of 1115 bytes, with an interleaver depth I equal to 5 and a block length, after R-S coding, equal to 1275 bytes. The structure of the TM transfer frames comprises several adjacent fields, which are represented in Fig. 10.

C. TM Encryption: Simulations and Results

To compare the performance of AES in CFB and OFB operational modes, with a special look at the problem of error propagation, we have simulated the transmission of 100 TM frames, each 1115 bytes long, by introducing, in any simulation, a number of errors ranging between 50 and 120. It is easy to see that this corresponds to having an average error rate over the channel on the order of 10^{-2}; the same analysis can be obviously repeated for different, and even more realistic, probabilities (i.e., 10^{-4}, 10^{-5}, or 10^{-6}). Part of the frame structure is fixed, and the values adopted are set according to [7]; in each case, the same 128-bit key and initialization vector, randomly selected, have been used. Finally, the content of the TM frame data field has been set equal to a randomly generated binary sequence.

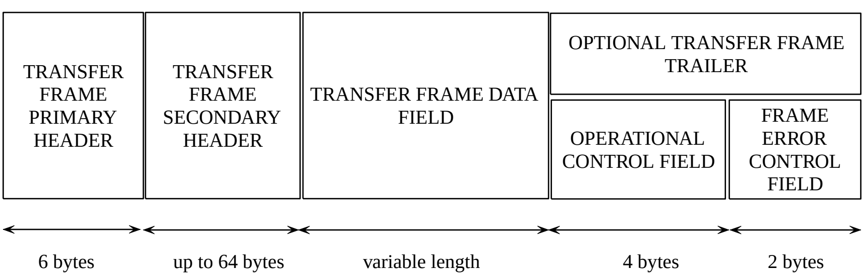


Fig. 10 TM transfer frame structure.

Two different situations have been considered: in the first one, noted by "AES . . . 1", the same TM frame, randomly generated once, has been transmitted for all of the simulations, while in the second one, noted by "AES. . . 100", the TM frame has been arbitrarily changed for any simulation. This choice depicts two different operation conditions that, we will show, do not produce necessarily the same result, particularly for the CFB operational mode.

Figure 11 shows the number of bit errors at the output of the decryption device as a function of the number of bit errors at the input, for the AES CFB and the AES OFB algorithms, by considering the two types of simulation described. The results confirm what is expected but, in addition, permit a more quantitative comparison between the two options. Looking at the figure, we see that AES encryption in OFB mode does not suffer from error propagation after decryption at the receiving side. In fact, both the curves "AES OFB 1" and "AES OFB 100" are straight lines, with unit slope, which means that the number of erred bits affecting the deciphered text is identical to the number of erred bits affecting the transmitted cipher text over the channel. The situation is quite different for the AES CFB operational mode, whose curves are highly irregular. In all cases, however, the number of errors out of the decryption device is, for such operational mode, much larger than the number of errors at the input, which implies that a propagation effect has occurred. The increase in the error rate is on the order of a factor of 25 for the "AES CFB 100" and a factor of 16 for the "AES CFB 1." This difference in the performance between the two simulated conditions also explains another expected behavior, that is the dependence of the propagation effect on the structure of the data. As a consequence, there is also a dependence on the position of the errors, at a parity of their number, as clearly testified by the observation that the "AES CFB" curves may be multivalued (the same number of input errors can yield different numbers of output errors).

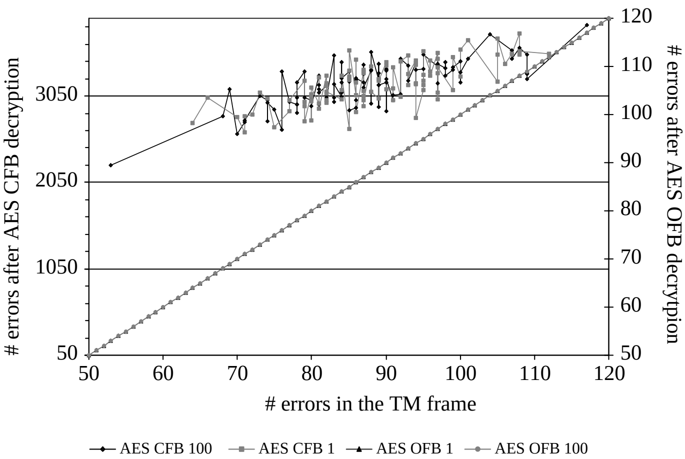


Fig. 11 AES CFB encryption of TM transfer frames: error rate comparison.

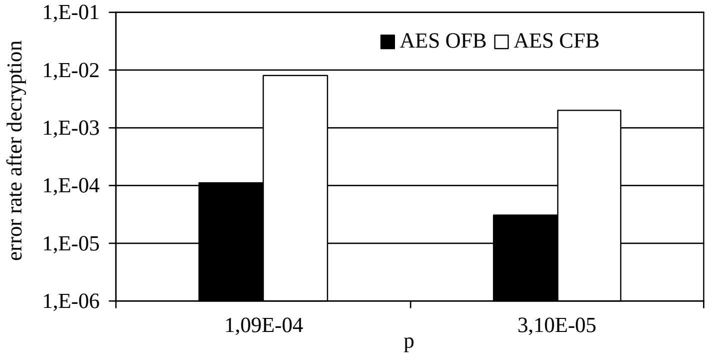


Fig. 12 AES OFB encryption of TM transfer frames: error rate comparison.

In Fig. 12 the simulation and comparison have been repeated by considering smaller values of the error probability p over the transmission channel. In this case, we have reported directly the average values resulting from a number of simulations (high enough to have a satisfactory confidence level). Previous conclusions are confirmed, as the AES OFB scheme performs better than the AES CFB, giving an error probability after decryption (expressed as the ratio between the number of erred bits after decryption and the total number of received bits) that is about two orders of magnitude lower, at a parity of the error probability over the channel. Actually, the error probability after decryption in the case of AES CFB is very high. This first series of results, however, refers to the case of absence of the error correcting code.

In a second round of simulations, we have introduced the R-S code specified in Sec III.B. We have considered channel error probabilities ranging between 10^{-7} and 10^{-4}. For these values, the R-S code is practically able to correct any combination of errors (the percentage of uncorrectable errors is quite negligible), and this is true when using either the CFB or the OFB scheme.

In the case of burst errors, performance depends on the length of the burst. For lengths smaller than 100 bits, the R-S code is able to correct, nearly completely, burst errors too. On the contrary, when the length exceeds 100 bits (we have considered values up to 1000 bits), this is no longer true, and the R-S code is not able to provide total correction. To compare the performance obtained when applying the CFB or the OFB encryption, in this more critical situation, we define a parameter, called *correction rate*, as the difference between the number of channel errors and the number of residual errors after decryption, normalized to the number of channel errors. If this value gets negative, it means that the error propagation effect due to the decryption process gives a number of errors after decryption that is higher than the number of errors affecting the channel, i.e., the correction properties of the R-S code are overcome by the side error propagation effect. Figure 13 reports, in percentage, the correction rate so defined in the case of CFB and OFB encryption. The correction rate can be referred to bits (bit correction rate) or to bytes (byte correction rate) or even to frames (frame correction rate); the

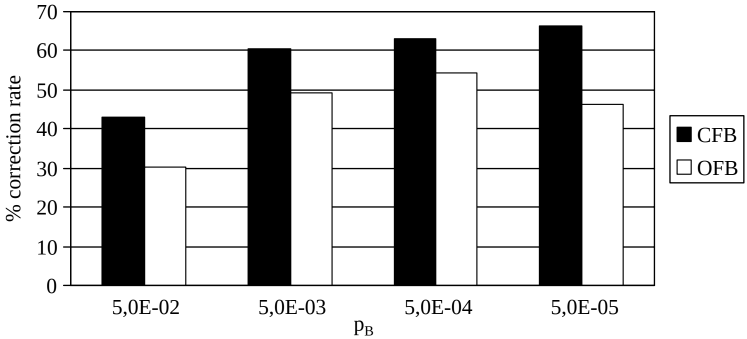


Fig. 13 Percent bit correction rate values in the case of CFB and OFB decryption.

results are not necessarily the same, also because of the byte-oriented nature of the R-S code. Though different from the channel error probability in the case of sparse errors, we can define an input error probability p_B also in this case as the average number of bit in error, because of bursts, over the total number of errors. For burst lengths between 2 and 1000, p_B varies between $5 \cdot 10^{-5}$ and $5 \cdot 10^{-2}$. Examples of bit, byte, and frame correction rate are given in Figs. 13–15 for some values of pB.

When the error rate on the channel overcomes the correction capacity of the R-S code, the two decryption schemes show different behaviors: at byte level, the OFB mode gives better performance than CFB (which is affected by an error propagation rate of 17 bytes, in our example), whereas CFB gives higher correction rate values at the bit level. We can say that the number of bits not correctly decrypted is higher in the OFB mode, but in the CFB mode they are spread on a higher number of bytes. Finally, we can point out the negative value of the percent byte correction rate, in the case of CFB mode, for $p_B = 5 \cdot 10^{-2}$: this means that

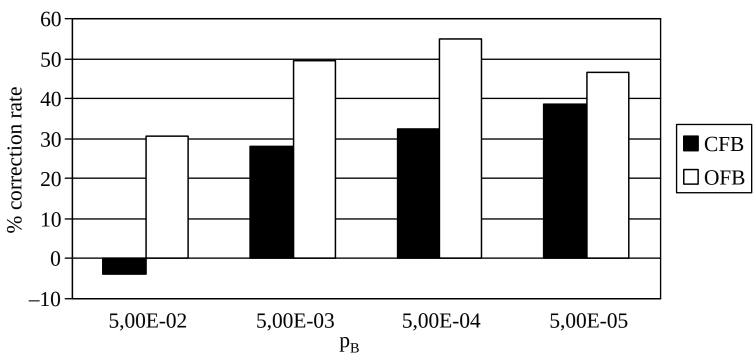


Fig. 14 Percent byte correction rate values in the case of CFB and OFB decryption.

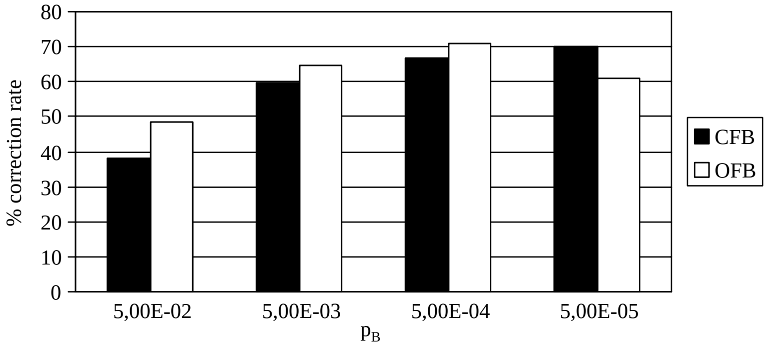


Fig. 15 Percent frame correction rate values in the case of CFB and OFB decryption.

the error propagation effect of the CFB decryption process is stronger than the correction capacity of the applied R-S code.

The results presented in this section seem to favor the adoption of the AES OFB mode for TM frames encryption; anyway, as previously stressed, other aspects should be considered and can become dominant for the choice. As an example, the OFB mode is known to be not as robust as the CFB mode against message stream modification attacks.

IV. Conclusion

This chapter has given a preliminary numerical evaluation for some security solutions applicable to telecommand and telemetry in space, with a particular emphasis on the schemes currently under investigation by the CCSDS Security WG. In space missions, we have typical needs for confidentiality, authentication, and integrity services. Integrity is a fundamental requirement for telecommands, to avoid modifications by malicious entities; besides that, auhentication should be supported to allow the verification of the TCs source. On the other hand, TM data usually require confidentiality, obtained by means of cryptographic transformations. In the case of TC authentication, we have compared the performance obtainable by the application of two different AES-based MAC schemes for signature generation, considering the effect of sparse errors on the transmission channel, by including or not the recommended error correction code. Additionally, concerning TM data protection, we have compared two different encryption schemes based on AES, drawing some preliminary conclusions on their possible adoption. Also in this case, the presence of forward error correction techniques implemented at the CCSDS data link layer has been taken into account, to face sparse and burst errors affecting the transmission channel.

The authentication and encryption techniques compared throughout this study do not show strong differences when tested under the same conditions. The most important features have been properly highlighted, but other aspects should be taken into account to evaluate their operational impact on space security systems design.

Therefore, discussion on these topics is in progress, and should benefit by numerical analyses like the one here presented. Further deepening of such an evaluation, by considering other options as well as more realistic scenarios, will be one of the targets of our future research.

References

[1] *CCDS Recommended Standard for Space Data Systems: The Application of CCSDS Protocols to Secure Systems*, Consultative Committee for Space Data Systems, CCSDS 350.0-G-2, Washington, DC, Jan. 2006.

[2] Weiss, H., "Authentication/Integrity Algorithm Issues White Paper," Consultative Committee for Space Data Systems, White Paper, Washington, DC, Aug. 2005; also http://public.ccsds.org/sites/cwe/sea-sec/default.aspx [retrieved 29 Nov. 2006].

[3] Weiss, H., *Encryption Algorithm Trade Study*, Consultative Committee for Space Data Systems, Washington, DC, July 2005; also http://public.ccsds.org/sites/cwe/sea-sec/default.aspx [retrieved 29 Nov. 2006].

[4] Spinsante, S., Baldi, M., Chiaraluce, F., Gambi, E., and Righi, G., "Evaluation of Authentication and Encryption Algorithms for Telecommand and Telemetry in Space Missions," *Proceedings of the 23rd AIAA International Communications Satellite Systems Conference*, [CD-ROM], Istituto Internazionale delle Comunicazioni, Rome, Italy, 25–28 Sept. 2005.

[5] "Packet Telecommand Standard," ESA PSS-04-107, Issue 2, Paris, April 1992.

[6] *CCDS Recommended Standard for Space Data Systems: TC Synchronization and Channel Coding*, Consultative Committee for Space Data Systems, CCSDS 231.0-R-1, Washington, DC, July 2002.

[7] *CCDS Recommended Standard for Space Data Systems: TM Synchronization and Channel Coding*, Consultative Committee for Space Data Systems, CCSDS 131.0-B-1, Washington, DC, Sept. 2003.

Chapter 23

Step Toward Interoperability for Spacecraft Monitoring and Control Automation Service

E. Poupart*

Centre National d'Etudes Spatiales, Toulouse, France

and

P. A. Cros†

Communication & Systèmes, Systèmes d'Informations (CS SI)
Toulouse, France

I. Introduction

IN THE context of the Spacecraft Monitoring and Control (SMC) working group of the Consultative Committee for Space Data Systems (CCSDS), the Centre National d'Etudes Spatiales (CNES) is responsible for the definition of the automation service in conformance with the service framework Mission Operations Services Concept described in the corresponding SMC Green Book [1]. This has been done in a research and development (RD) study that started in December 2005 and ended in April 2006. The results of this study will be described in this chapter. This study was done with the following objectives:

1) Defining formally automation to guarantee that the process will execute exactly as specified.

2) Identifying SMC services involved and validate their completeness.

3) Improve interoperability for a better cooperation between different agencies and contractors.

II. Automation Service Context

Automation in ground systems is crucial to have efficient and cost-effective spacecraft operations. Automation is achieved when scripted procedures are used

---

*Software Engineer, Products & Ground System Directorate/Generic Ground Systems Office.

†Software Engineer, Space Department.

to conduct normal operations, such as configuration of ground equipment for a satellite contact, commanding the spacecraft or payload to a new configuration, commanding a spacecraft maneuver, and monitoring of spacecraft data. Experience has shown that these procedures substantially reduce both operator workload and the potential for errors during testing and mission operations.

These scripted procedures involve commands to the spacecraft, commands to ground equipment, references to ground equipment status, references to current telemetry values, and references to local data. Some structured programming constructs are usually supported, although the language structure is often kept at an operations level rather than attempting to utilize conventional programming languages.

Different ground systems have implemented automation using various languages, including some visual programming languages. Typically these languages have been proprietary and incompatible between different ground system developers and vendors. Transfer of a satellite from one ground system to another ground system, as would occur during a ground system upgrade [or electric ground support equipment (EGSE) to control center], is therefore more expensive due to the required conversion of thousands of lines of automation procedure code.

In the CCSDS SMC working group, we are currently defining three low-level services: one called *Core Service* to monitor and check parameters and to send commands, one called *Common Service* to manage services, interaction between services, persistency and security, and one called *Spacecraft Monitoring and Control Protocol Service* to manage protocols. Three high-priority standards corresponding to those services are currently in definition, and after a validation through prototyping, they should become CCSDS Recommended Standards.

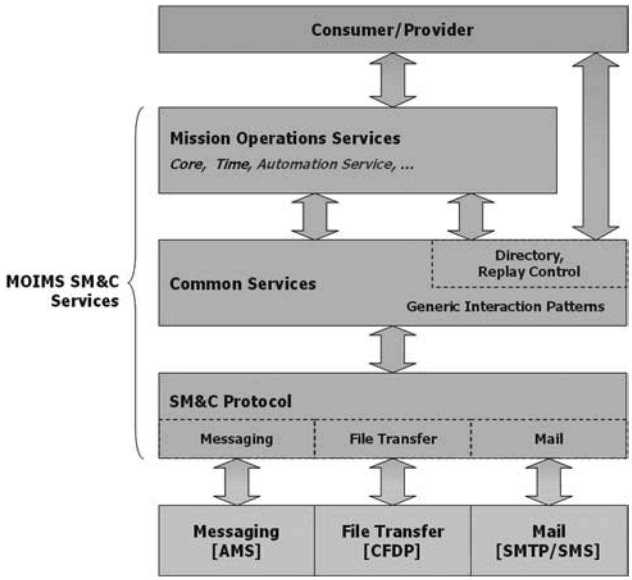


Fig. 1 MOIMS SMC services.

Automation service, which is a higher level service, should use those services and also other high-level ones currently in definition like *Time Service* and *Remote Software Management*.

Automation process is usually described at design time through a *workflow specification* (called *procedure* in the space domain) using a dedicated language. Workflow execution corresponds to a path in a graph (workflow specification) from an initial state to a final state as described in Fig. 2.

To be able to say what and how automation service could be provided by the automation process, it was necessary to study it in the space domain and identify its goals. This was done in European spacecraft operations languages and tools (PLUTO, ELISA, MOIS, TMTCS) and their context.

To identify which SMC services were involved and their completeness, it was necessary to identify them in existing procedures coming from different contexts. This was done on procedures operating DEMETER and PARASOL microsatellites, ATV, ARIANE, and Telecom 2 satellite. Operational people involved in EGSE or different control centers were interviewed.

One important key point of this study was to make a clear separation between the concepts related to 1) the automation process description; 2) the services used by this automation process (defined in current SMC red books); 3) the services using this automation process (automation, scheduling, etc.); and 4) the automation service itself. Like this, the automation process engine only takes care of the "What" and "When" (Automation process described in Sec. III) while the SMC services take care of the "How" and "By whom" (described in Sec. IV).

Another key point to achieve interoperability is the formal semantic about concepts to precisely how they should be interpreted, because ambiguous definitions lead to misunderstanding and make automation components unable to "understand" each other. Automation interoperability may be achieved if the concepts concerning all the parts (SMC services and automation process) are defined formally.

Concerning automation process, in the space domain, there are many spacecraft operations languages, not only in Europe, which contain concepts related to it. Also, outside the space domain, the situation is similar with many automation languages, also called *workflow* languages, unable to understand each other.

Many studies had already been made at Queensland University of Technology, Australia (QUT) and Eindhoven University of Technology, Netherlands (EUT) on more than 30 existing workflow systems and languages with standards like XPDL,

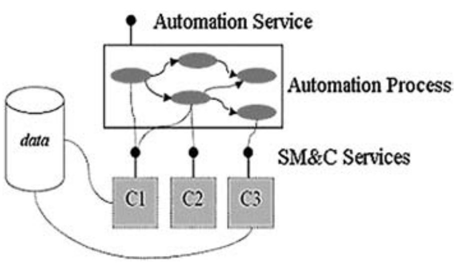


Fig. 2 Automation process.

BPEL4WS, XLANG, UML2 AD, etc., focusing on differences and interoperability aspects. To achieve those studies, they used workflow patterns related to control flow, data, and resource perspectives. The conclusion was that all workflow patterns were not supported and that there was a lack of semantic description.

The Yet Another Workflow Language (YAWL) was produced based on Petri net formalism to solve this problem. We have done a similar study using those patterns on Spacecraft Operations European languages and tools like PLUTO, ELISA, TMTCS and MOIS. The results are described in the following sections.

III. Automation Process Concepts

A. Use of Petri Net Formalism to Achieve Automation Process Interoperability

1. Similar Context Outside CCSDS Context [2]

In the area of workflow, one is confronted with a plethora of products (commercial, free, and open source) supporting languages that differ significantly in terms of concepts, constructs, and their semantics. One of the contributing factors to this problem is the lack of a commonly agreed upon formal foundation for workflow languages. Standardization efforts, e.g., XPDL [3] proposed by the WFMC, have essentially failed to gain universal acceptance and have not in any case provided such a formal basis for workflow specification. The lack of well-grounded standards in this area has induced several issues, including minimal support for migration of workflow specifications, potential for errors in specifications due to ambiguities, and lack of a reference framework for comparing the relative expressive power of different languages.

2. Formal Semantic Based on Petri Nets

While workflow patterns provide a pragmatic approach to control flow specification in workflows, Petri nets provide a more theoretical approach. Petri nets [4] form a model for concurrency with a formal foundation, an associated graphical representation, and a collection of analysis techniques. These features, together with their direct support for the notion of state (required in some of the workflow patterns), makes them attractive as a foundation for control flow specification in workflows. However, even though Petri nets (including higher order Petri nets such as colored petri nets [4]) support a number of the identified patterns, they do not provide direct support for the cancellation patterns (in particular the cancellation of a whole case or a region), the synchronizing merge pattern (where all active threads need to be merged, and branches that cannot become active need to be ignored), and patterns dealing with multiple active instances of the same activity in the same case [5]. This realization motivated the development of YAWL [6], which combines the insights gained from the workflow patterns with the benefits of Petri nets. It should be noted, though, that YAWL is not simply a set of macros defined on top of Petri nets. Its semantics are not defined in terms of Petri nets but rather in terms of a transition system.

As a language for the specification of control flow in workflows, YAWL has the benefit of being highly expressive and suitable, in the sense that it provides

direct support for all the workflow patterns, while the reviewed workflow languages provide direct support for only a subset of them. The expressive power and formal semantics of YAWL make it an attractive candidate to be used as an intermediate language to support translations of workflows specified in different languages. The YAWL language is based on an analysis of more than 30 workflow systems, languages, and standards with UML2 AD.

YAWL [6] is based on a line of research grounded in Petri net theory [7, 4] and the 20 workflow patterns documented in [8]. Analyses of BPMI's BPML [9] and WFMC's XPDL [3] using the patterns are also available via www.workflowpatterns.com. In total, more than 30 languages/systems have been evaluated, and these evaluations have driven the development of the YAWL language.

Workflow specifications can be understood, in a broad sense, from a number of different perspectives (see [10], [11]):

1) The control flow (or process) perspective describes activities and their execution ordering through different constructors, which permit flow of execution control, e.g., sequence, choice, parallelism, and synchronization. Activities in elementary form are atomic units of work, and in compound form modularize an execution order of a set of activities.

2) The data perspective layers business and processing data on the control flow perspective. Data that flow between activities, and local variables of the workflow, qualify in effect pre- and post-conditions of activity execution.

3) The resource perspective provides an organizational structure anchor to the workflow in the form of human and device roles responsible for executing activities.

4) The operational perspective describes the elementary actions executed by activities, where the actions map into underlying applications. Typically, (references to) workflow data are passed into and out of applications through activity-to-application interfaces, allowing manipulation of the data within applications.

Clearly, the control flow perspective provides an essential insight into a workflow language's effectiveness. The dataflow perspective rests on it, while the organizational and operational perspectives are ancillary.

3. *Workflow Terminology*

The primary task of a workflow management system [11] is to enact case-driven business processes by allowing workflow models to be specified, executed, and monitored.

Workflow specifications (workflow schemas) are defined to specify which activities need to be executed and in what order (i.e., the routing or control flow).

An *elementary activity* is an atomic piece of work.

Workflow specifications are instantiated for specific *cases* (i.e., *workflow instances*). Examples of business cases in the space domain are onboard sampling change and disabling limit checks before instrument maneuver, mass memory data acquisition and checking of anomalies data, putting ON or OFF equipments, etc. Since a case is an instantiation of a process definition, it corresponds to the execution of concrete work.

Activities are connected through transitions and we use the notion of a *thread of execution control* for *concurrent executions* in a workflow context. Activities

are undertaken by roles that define organizational entities, such as humans and devices.

Control data are data introduced solely for workflow management purposes, e.g., variables introduced for routing purposes.

Elementary actions are performed by roles while executing an activity for a specific case, and are executed using applications (any external application or component implementing a service to be activated).

In the vast majority of workflow management systems [12] when an activity instance is finished, the next activity instance to be executed is selected and its state is changed to READY (this typically corresponds to placing it on a designated work list). After this, the activity instance can go through a number of internal states. Finally, if all the associated processing has been performed successfully, its state is changed to COMPLETE. These two states are crucial to control flow considerations and any formal semantics of control flow constructs has to take at least these two states into account explicitly.

4. *Data Characterization*

From a data perspective [13], there are a series of characteristics that occur repeatedly in different workflow modeling paradigms. These can be divided into four distinct groups:

1) Data visibility: relating to the extent and manner in which data elements can be viewed by various components of a workflow process.

2) Data interaction: focusing on the manner in which data are communicated between active elements within a workflow.

3) Data transfer: considers the means by which the actual transfer of data elements occurs between workflow components and describes the various mechanisms by which data elements can be passed across the interface of a workflow component.

4) Data-based routing: characterizes the manner in which data elements can influence the operation of other aspects of the workflow, particularly the control flow perspective.

5. *Workflow Structure*

A workflow model is composed of a number of tasks that are connected in the form of a directed graph. An executing instance of a workflow model is called a case or process instance. There may be multiple cases of a particular workflow model running simultaneously, however, each of these is assumed to have an independent existence, and they typically execute without reference to each other.

There is usually a unique first task and a unique final task in a workflow. These are the tasks that are first to run and last to run in a given workflow case. Each invocation of a task that executes is termed a task instance. A task instance may initiate one or several task instances when it completes. This is illustrated by an arrow from the completing task to the task being initiated, e.g., in Fig. 3, task instance B is initiated when task instance A completes.

This may also occur conditionally and where this is the case, the edge between task instances indicates the condition that must be satisfied for the subsequent task

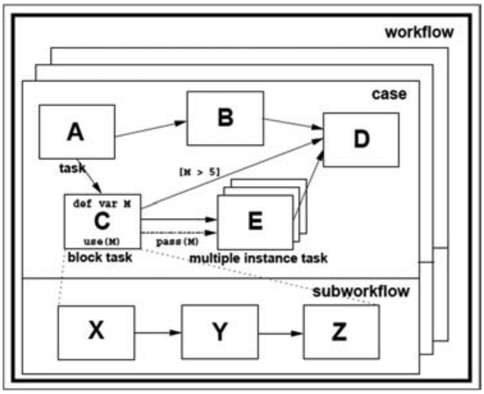


Fig. 3 Components of a workflow [13].

instance to be started, e.g., task instance D is initiated when task instance C completes if the data element M is greater than 5.

A task corresponds to a single unit of work. Four distinct types of task are denoted: atomic, block, multi-instance and multi-instance block. We use the generic term components of a workflow to refer to all of the tasks that comprise a given workflow model.

An atomic task is one that has a simple, self-contained definition (i.e., one that is not described in terms of other workflow tasks) and only one instance of the task executes when it is initiated.

A block task is a complex action that has its implementation described in terms of a sub-workflow. When a block task is started, it passes control to the first task(s) in its corresponding sub-workflow. This sub-workflow executes to completion and at its conclusion, it passes control back to the block task, e.g., block task C is defined in terms of the sub-workflow comprising tasks X, Y, and Z.

A multiple-instance task is a task that may have multiple distinct execution instances running concurrently within the same workflow case. Each of these instances executes independently. Only when a nominated number of these instances have completed is the task following the multiple instance task initiated.

A multi-instance block task is a combination of the two previous constructs and denotes a task that may have multiple distinct execution instances, each of which is block structured in nature (i.e., has a corresponding sub-workflow).

B. Control Flow Patterns Identified in Spacecraft Operations Languages

1. *Basic Control Flow Patterns [11]*

The basic control flow patterns (see Fig. 4) are as follows:

1) *Pattern 1 (Sequence)*

Description: An activity in a workflow process is enabled after the completion of another activity in the same process.

Synonyms: Sequential routing, serial routing.

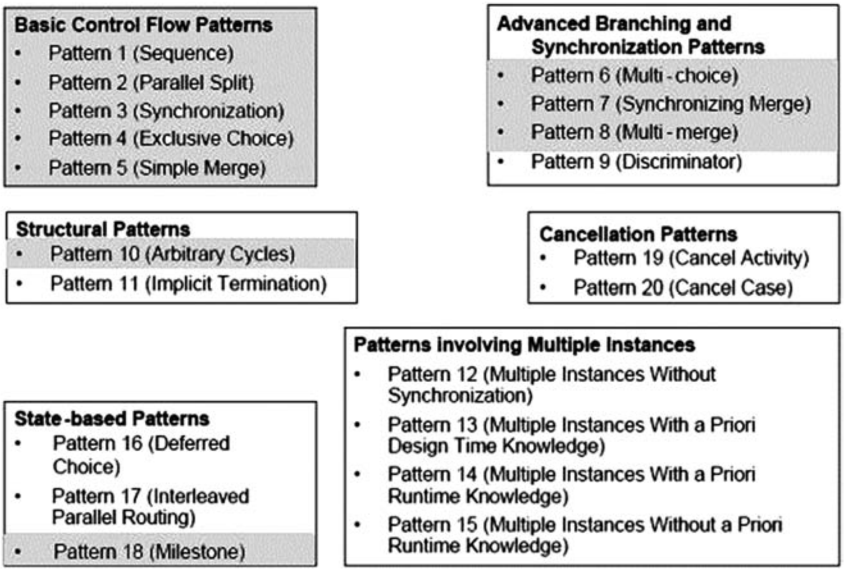


Fig. 4 Control flow patterns.

2) *Pattern 2 (Parallel Split)*

Description: A point in the workflow process where a single thread of control splits into multiple threads of control that can be executed in parallel, thus allowing activities to be executed simultaneously or in any order.

Synonyms: AND-split, parallel routing, fork.

3) *Pattern 3 (Synchronization)*

Description: A point in the workflow process where multiple parallel subprocesses/activities converge into one single thread of control, thus synchronizing multiple threads. It is an assumption of this pattern that each incoming branch of a synchronizer is executed only once.

Synonyms: AND-join, rendez-vous, synchronizer.

4) *Pattern 4 (Exclusive Choice)*

Description: A point in the workflow process where, based on a decision or workflow control data, one of several branches is chosen.

Synonyms: XOR-split, conditional routing, switch, decision.

5) *Pattern 5 (Simple Merge)*

Description: A point in the workflow process where two or more alternative branches come together without synchronization. It is an assumption of this pattern that none of the alternative branches is ever executed in parallel (if this is not the case, then see Pattern 8 (Multi-merge) or Pattern 9 (Discriminator)).

Synonyms: XOR-join, asynchronous join, merge.

2. Advanced Branching and Synchronization Patterns

These patterns do not have straightforward support in most workflow engines. Nevertheless, they are quite common in real-life business scenarios:

1) *Pattern 6 (Multi-choice)*

Description: A point in the workflow process where, based on a decision or workflow control data, a number of branches are chosen.

Synonyms: Conditional routing, selection, OR-split.

2) *Pattern 7 (Synchronizing Merge)*

Description: A point in the workflow process where multiple paths converge into one single thread. If more than one path is taken, synchronization of the active threads needs to take place. If only one path is taken, the alternative branches should reconverge without synchronization. It is an assumption of this pattern that a branch that has already been activated, cannot be activated again while the merge is still waiting for other branches to complete.

Synonyms: Synchronizing join.

Problem: The main difficulty with this pattern is to decide when to synchronize and when to merge. Generally speaking, this type of merge needs to have some capacity to be able to determine whether it may (still) expect activation from some of its branches.

3) *Pattern 8 (Multi-merge)*

Description: A point in a workflow process where two or more branches reconverge without synchronization. If more than one branch gets activated, possibly concurrently, the activity following the merge is started for every activation of every incoming branch.

Synonyms: None.

3. *Structural Patterns [11]*

Different workflow management systems impose different restrictions on their workflow models. These restrictions (e.g., arbitrary loops are not allowed, only one final node should be present, etc.) are not always natural from a modeling point of view and tend to restrict the specification freedom of the business analyst. As a result, business analysts either have to conform to the restrictions of the workflow language from the start, or they model their problems freely and transform the resulting specifications afterwards. A real issue here is that of suitability. In many cases the resulting workflows may be unnecessarily complex, which impacts end-users who may wish to monitor the progress of their workflows. In this section two patterns are presented that illustrate typical restrictions imposed on workflow specifications and their consequences.

Virtually every workflow engine has constructs that support the modeling of loops. Some of the workflow engines provide support only for what we will refer to as structured cycles. Structured cycles can have only one entry point to the loop and one exit point from the loop and they cannot be interleaved. They can be compared to WHILE loops in programming languages while arbitrary cycles are more like GOTO statements. This analogy should not deceive the reader, though, into thinking that arbitrary cycles are not desirable, as there are two important differences here with classical programming languages: 1) the presence of parallelism, which in some cases makes it impossible to remove certain forms of arbitrariness; and 2) the fact that the removal of arbitrary cycles may lead to workflows that are much harder to interpret (and as opposed to programs, workflow specifications also have to be understood at runtime by their users).

One structural pattern identified in spacecraft operations language is:

1) *Pattern 10 (Arbitrary Cycles)*

Description: A point in a workflow process where one or more activities can be done repeatedly.

Synonyms: Loop, iteration, cycle.

4. State-Based Patterns [11]

In real workflows, most workflow instances are in a state awaiting processing rather than being processed. Moments of choice, such as supported by constructs as XOR-splits/OR-splits, in workflow management systems are typically of an explicit nature, i.e., they are based on data or they are captured through decision activities. This means that the choice is made a priori, i.e., before the actual execution of the selected branch starts an internal choice is made. Sometimes this notion is not appropriate. The state-based pattern is the following:

1) *Pattern 18 (Milestone)*

Description: The enabling of an activity depends on the case being in a specified state, i.e., the activity is only enabled if a certain milestone has been reached that did not expire yet. Consider three activities named A, B, and C. Activity A is only enabled if activity B has been executed and C has not been executed yet, i.e., A is not enabled before the execution of B and A is not enabled after the execution of C. The state in between B and C is modeled by place M. This place is a milestone for A. Note that A does not remove the token from M: it only tests the presence of a token.

Synonyms: Test arc, deadline, state condition, withdraw message.

Problem: There is a race between a number of activities and the execution of some activities may disable others. In most workflow systems (notable exceptions are those based on Petri nets), once an activity becomes enabled, there is no other-than-programmatic way to disable it. A milestone can be used to test whether some part of the process is in a given state. Simple message passing mechanisms will not be able to support this because the disabling of a milestone corresponds to withdrawing a message.

C. Dataflow Patterns Identified in Spacecraft Operations Languages

The dataflow patterns are illustrated in Fig. 5.

1. Data Visibility [13]

The data visibility patterns identified in spacecraft operations languages are as follows:

1) *Pattern 1 (Task Data)*

Description: Data elements can be defined by tasks that are accessible only within the context of individual execution instances of that task.

2) *Pattern 5 (Case Data)*

Description: Data elements are supported that are specific to a process instance or case of a workflow. They can be accessed by all components of the workflow during the execution of the case.

Nr	Pattern
1	Task Data
2	Block Data
3	Scope Data
4	Multiple Instance Data
5	Case Data
6	Workflow Data
7	Environment Data

Nr	Pattern
8	Data Interaction between Tasks
9	Data Interaction – Block Task to Sub-workflow Decomposition
10	Data Interaction – Sub-workflow Decomposition to Block Task
11	Data Interaction – to Multiple Instance Task
12	Data Interaction – from Multiple Instance Task
13	Data Interaction – Case to Case

Nr	Pattern
14	Data Interaction – Task to Environment – Push-Oriented
15	Data Interaction – Environment to Task – Pull-Oriented
16	Data Interaction – Environment to Task – Push-Oriented
17	Data Interaction – Task to Environment – Pull-Oriented
18	Data Interaction – Case to Environment – Push-Oriented
19	Data Interaction – Environment to Case – Pull-Oriented
20	Data Interaction – Environment to Case – Push-Oriented
21	Data Interaction – Case to Environment – Pull-Oriented
22	Data Interaction – Workflow to Environment – Push-Oriented
23	Data Interaction – Environment to Workflow – Pull-Oriented
24	Data Interaction – Environment to Workflow – Push-Oriented
25	Data Interaction – Workflow to Environment – Pull-Oriented

Nr	Pattern
33	Task Precondition – Data Existence
34	Task Precondition – Data Value
35	Task Postcondition – Data Existence
36	Task Postcondition – Data Value
37	Event-based Task Trigger
38	Data-based Task Trigger
39	Data-based Routing

Fig. 5 Dataflow patterns.

3) *Pattern 6 (Workflow Data)*

Description: Data elements are supported that are accessible to all components in each and every case of the workflow and are within the control of the workflow system.

To deal with data (nested expressions, etc.), most of the existing workflow management systems use a propriety language for dealing with data. YAWL is one of the few languages that completely relies on XML-based standards like XPath and XQuery.

2. Data Interaction

Data elements can be passed between components in a workflow process and how the characteristics of the individual components can influence the manner in which the trafficking of data elements occurs. Of particular interest is the distinction between the communication of data between components within a workflow engine as against the data-oriented interaction of a workflow element with the external environment.

Internal data interaction involves the following patterns:

1) *Pattern 9 [Data Interaction (Block Task to Sub-Workflow Decomposition)]*

Description: The ability to pass data elements from a block task instance to the corresponding sub-workflow that defines its implementation.

2) *Pattern 10 [Data Interaction (Sub-Workflow Decomposition to Block Task)]*

Description: The ability to pass data elements from the underlying sub-workflow back to the corresponding block task instance.

External data passing involves the communication of data between a component of a workflow process and some form of information resource or service that is operated outside of the context of the workflow engine (see Fig. 6). The notion of being external to the context of the workflow engine applies not only in technology terms but also implies that the operation of the external service or resource is independent of that of the workflow engine. The patterns are as follows:

1) *Pattern 14 [Data Interaction (Task to Environment, Push-Oriented)]*

Description: The ability of a task to initiate the passing of data elements to a resource or service in the operating environment.

2) *Pattern 15 [Data Interaction (Environment to Task, Pull-Oriented)]*

Description: The ability of a workflow task to request data elements from resources or services in the operational environment.

3. Data-Based Routing

The following patterns capture the various ways in which data elements can interact with other perspectives and influence the overall operation of the workflow:

1) *Pattern 34 [Task Precondition (Data Value)]*

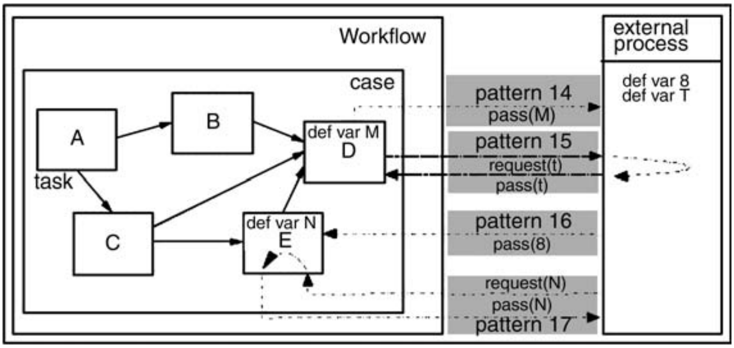


Fig. 6 External data interaction between workflow and environment [13].

Description: Data-based preconditions can be specified for tasks based on the value of specific parameters at the time of execution.

For automation, other possibilities should be added when task precondition is not met, like ask the operator, suspend the task, stop the task, etc.

2) *Pattern 36 [Task Post-Condition (Data Value)]*

Description: Data-based post-conditions can be specified for tasks based on the value of specific parameters at the time of execution.

Motivation: Implementation of this pattern would ensure that a task could not complete until specific output parameters had a particular data value or are in a specified range.

Implementation: Two options exist for handling the achievement of specified values for data elements at task completion: 1) delay execution until the required values are achieved, 2) implicitly re-run the task.

For automation, other possibilities should be added like ask the operator, suspend the task, stop the task, etc.

A default task post-condition handler should exist for every task and is defined as follows: If the execution result of the task is not okay, then the task should be suspended.

3) *Pattern 37 (Event-Based Task Trigger)*

Description: The ability for an external event to initiate a task (see Fig. 7).

4) *Pattern 38 (Data-Based Task Trigger)*

Description: The ability to trigger a specific task when an expression based on workflow data elements evaluates to true (see Fig. 8).

5) *Pattern 39 (Data-Based Routing)*

Description: The ability to alter the control flow within a workflow case as a consequence of the value of data-based expressions (see Fig. 9).

D. Missing Concepts in the Preceding Patterns

Error handling is a missing concept in the preceding patterns. Concerning error handling, there is nothing for the timebeing in the control-flow patterns, but it

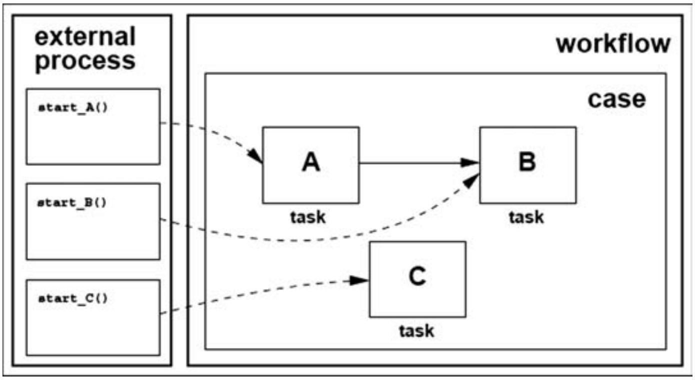


Fig. 7 Event-based task trigger [13].

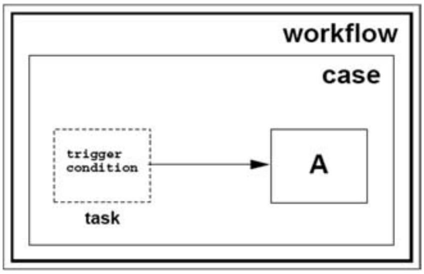


Fig. 8 Data-based task trigger.

should come in the near future! This concept is important for SMC automation, which needs to have the capability to associate an error handler to each activity and task to suspend it, and start a contingency activity for example.

Concerning automation activity, it must always have one default handler that must suspend it and display the error message. This default error handler may be overridden with a specific one to suspend the current activity in error, start another activity (contingency procedure), then after the successful end of the contingency activity, resume the activity in error for example.

There is one concept in Unified Modeling Language-2 (UML2) Activity Diagrams that could be a good starting point to describe error handling.

The structured exception handling facility from UML2 AD is analogous to try/throw/catch constructs in programming languages. It provides a way to indicate that a structured node or action traps exceptions raised from inside it or from behaviors it calls. An exception in UML is any object thrown with the predefined action RAISEEXCEPTIONACTION. It assumes the CHECKORDER behavior will raise an exception of type NOFILLREASON if the order does not pass the

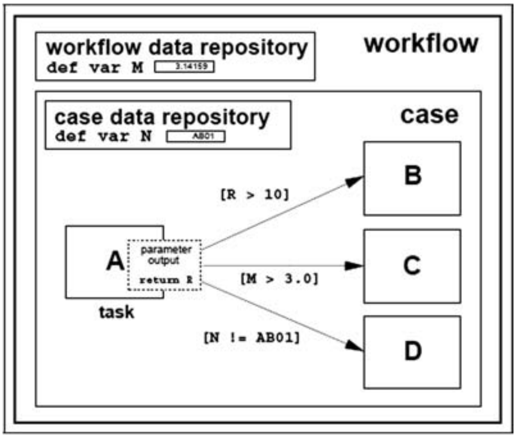


Fig. 9 Data-based routing [13].

check. When this happens, all tokens flowing in the execution of CHECKORDER and the node invoking it are destroyed. The zigzag arrow to the input pin of NOTIFYBUYER indicates the structured node traps exceptions of type NOFILLREASON, and the reaction will be to notify the buyer that something is wrong with the order. In general, an exception is passed from the point at which it is raised, up the "call tree" through all containing structured nodes, activities, synchronous call behavior, and operation actions, until it reaches a structured node or action protected by an exception handler for the type of exception raised. All tokens are destroyed in constructs the exception object passes through on the way up to the handler. Then the handler is run. UML does not specify what happens if no handler is found for an exception at all.

Exceptions are not passed up through asynchronous invocation actions. These actions do not expect a reply, and separate the caller and callee completely.

IV. SMC Services Involved in Automation

Concerning the services executed by Spacecraft Operations languages, we studied existing procedures operating DEMETER and PARASOL microsatellites, ARIANE, ATV, and the Telecom 2 satellite. Then, we mapped those services on SMC services. The results are described next.

A. SMC Services Used Externally by SMC Automation Service

The SMC core services used (see Table 1) are the following:

1) SMC Core.Monitoring.Status (register, deregister, and update) service to perform checks on telemetry parameters.

2) SMC Core.Monitoring.behaviour service to enable/disable parameter checks.

3) SMC Core.Monitoring.Action and Core.Monitoring.Verification service to send commands to the spacecraft.

4) Core.Monitoring.Alerts service to get alerts from the spacecraft or from the ground monitoring activity.

The SMC common services used (see Table 2) are the following:

Table 1 SMC core services used by automation

Core area	Capability set
Monitoring	Status Behavior Statistics Aggregation
Alerts	Alerts
Actions	Action Verification
Conversion	Engineering unit conversion

1) SMC Common.Active.Observe, Common.Active.Interact and Common. History services to interact with the operator, log information, send or receive mission specific events.

The SMC higher level services used are the following:

1) Automation.Control service itself to stop/start or suspend/resume the current activity.

2) Automation Service onboard to send delayed commands to the spacecraft or onboard procedures.

3) Time Service to get timeout events, set timeout in a relative or absolute way. This service under definition should be enriched to add those capabilities.

4) Flight dynamics service to get orbit, attitude, etc.

5) Remote Software Management service to load software image onboard, etc.

B. SMC Services Used Internally by SMC Automation to Provide Its Service

The main interactions between the automation engine and SMC services are bilateral with 1) the capability to start an operation on a service using a request/response pattern, 2) the capability to send/receive events to/from services, and 3) the capability to throw/catch errors to/from services. These interactions should have the possibility to sit on many protocols.

To be able to interact with SMC services, automation service should also provide the capability to register/unregister them into a directory service.

All of these capabilities should be provided by SMC Common services.

The capability to throw/catch errors is missing in the current release of SMC services.

C. SMC Services Provided by SMC Automation

The capabilities defined in the current release of SMC Automation service are as shown in Table 3.

Table 2 SMC common services used by automation

Common area	Functional area
Active	Observe
	Manage
	Control
	Interact
	Login
History	History management
	Replay Control
	Retrieve
Configuration	Configure
	Define
Directory	Directory

Table 3 SMC automation services provided by automation

Automation service area	Capability set
Control	Load
	Start
	Abort
	Suspend
	Resume
	Run step by step
	Manage break points
Monitoring	Register interest
	Deregister interest
	List available activities
	List executing activities
	Update activity status
Configuration	List procedure definitions
	Load procedure definition
	Dump procedure definition

V. Conclusion

The main results of this study are as follows:

1) It is possible to define formally the automation process using Petri nets.

2) Interoperability is possible at the automation process level between different Spacecraft Operation Language engines.

3) Automation service fits well into CCSDS SMC architecture.

Some further studies could be made concerning automation process specification at design time, using a higher level easy to understand for the operation specialist and about code generation into the engine dedicated language.

Acknowledgments

Thanks to Lachlan Aldred from Queensland University of Technology, Co-Director BPM Center, and developer of the Yawl engine, for his availability for questions on Yawl.

References

[1] *CCDS Recommended Standard for Space Data Systems: Mission Operations Services Concept*, Consultative Committee for Space Data Systems, CCSDS 520.0-G-2, Washington, DC, Aug. 2006.

[2] Van der Aalst, W. M. P., Aldred, L., Dumas, M., ter Hofstede, A. H. "Design and Implementation of the YAWL System."

[3] "Workflow Management Coalition Workflow Standard: Workflow Process Definition Interface – XML Process Definition Language (XPDL) (WFMC-TC-1025)," Workflow Management Coalition, Technical Rept., Lighthouse Point, FL, 2002.

[4] Jensen, K., *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*, EATCS monographs on Theoretical Computer Science, Springer-Verlag, Berlin, 1992.

[5] van der Aalst, W. M. P., and ter Hofstede, A. H. M., "Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages," *Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002)*, edited by K. Jensen, Vol. 560, Univ. of Aarhus, Aarhus, Denmark, 2002. pp. 1–20.

[6] van der Aalst, W. M. P., and ter Hofstede, A. H. M., "YAWL: Yet Another Workflow Language," Queensland Univ. of Technology, QUT Technical Rept. FIT-TR-2003-04, Brisbane, Australia, 2003, also *Information System* (to be published).

[7] van der Aalst, W. M. P., "The Application of Petri Nets to Workflow Management," *The Journal of Circuits, Systems and Computers*, Vol. 8, No. 1, 1998, pp. 21–66.

[8] van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., and Barros, A. P., "Workflow Patterns," *Distributed and Parallel Databases*, Vol. 14, No. 1, 2003, pp. 5–51.

[9] Arkin, A., et al., Business Process Modeling Language (BPML), Version 1.0, 2002.

[10] Jablonski, S., and Bussler, C., Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, London, UK, 1996.

[11] van der Aalst, W. M. P., ter Hofstede, A. H. M., Kiepuszewski, B., and Barros, A. P., "Workflow Patterns."

[12] Kiepuszewski, B., ter Hofstede, A. H. M., and van der Aalst, W. M. P., "Fundamentals of Control Flow in Workflows."

[13] Russell, N., ter Hofstede, A. H. M., Edmond, D., and van der Aalst, W. M. P., "Workflow Data Patterns."

VII. Earth-Orbiting Missions

This page intentionally left blank

Chapter 24

PARASOL and CALIPSO: Experience Feedback on Operations of Micro- and Small Satellites

Fabienne Serène* and Nathalie Corcoral†

Centre National d'Etudes Spatiales, Centre Spatial de Toulouse
Toulouse, France

I. Introduction

PROTEUS and MYRIADE are two satellites with low-cost product lines developed by the Centre National d'Etudes Spatiales (CNES), the first one being a small satellite family, the second one a microsatellite family. PARASOL, launched in December 2004, is the second MYRIADE microsatellite and is operated from CNES Toulouse Space Center (CST). The Cloud Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO), from the PROTEUS product line, launched in April 2006, is also operated from CST. PARASOL and CALIPSO are part of the Afternoon Train (A-Train), which is a constellation of six satellites coordinated by the Constellation Coordination System (NASA Goddard Space Flight Center).

CALIPSO is a three-year Earth-science Franco–American (CNES/NASA) mission. Its purpose is to study the clouds and aerosols radiative impacts that represent the main uncertainties about the climate evolution prediction. CALIPSO, as well as PARASOL, has integrated the A-Train satellite constellation with an orbit altitude of 705 km, and a nominal inclination of approximately 98.2 deg, local time being around 13:30 UTC.

The PARASOL mission purpose is to perform measurements of the polarized and multi-directional reflectances, on ground areas previously observed by the Calipso light detection and ranging (LIDAR).

The "LOA-CNRS" (Atmospheric Optics Laboratory–Centre National de la Recherche Scientifique) in the French city of Lille is the Parasol scientific principal investigator. For CALIPSO, there are two principal investigators, one

*PARASOL Operation Manager, Micro and Small Satellites Operations Office.

†CALIPSO Operation Manager, Micro and Small Satellites Operations Office.

American, the NASA Langley Research Center, located in Hampton, Virginia, the second located in the French city of Jussieu, the Pierre Simon Laplace Institut (IPSL/SA).

This chapter will first present PARASOL and CALIPSO satellites and operations, will present the combined station-keeping operations, will briefly describe the ground segment, and will present the chosen organization to allow the good execution of all activities (manpower, planning, coordination meetings, resources sharing, and operations priorities management).

Finally, the chapter will conclude with the presentation of future missions for both product lines and their integration into the operational system, taking into account the experience feedback acquired during the operations of small and microsatellites.

II. Satellite Description

A. PARASOL Microsatellite

The PARASOL satellite, launched 18 December 2004, stands about 80 cm tall and weighs 120 kg at launch. The satellite payload consists of a digital camera with a 274 × 242-pixel CCD detector array, wide-field telecentric optics, and a rotating filter wheel enabling measurements at different wavelengths and in several polarization directions. Because it acquires a sequence of images every 20s, the instrument can view ground targets from different angles.

The satellite attitude control system, precise within one-tenth of a degree, is built around a star sensor, four reaction wheels, and three magnetic torkers. Three sun sensors and a magnetometer are also used during the satellite positioning phase. The propulsion module is a blow down system using 4.5 kg of hydrazine, which corresponds to a speed increment of 85 m/s. It uses four thrusters, 1-N thrust each. The solar array provides approximately 180 W of electric power at the satellite's start of life. A lithium-ion battery provides power during eclipses. Onboard data handling is centralised and controlled by a 10-Mips T805 microprocessor. Data can be stored in a large mass memory (16 Gbits) and has a high-speed telemetry system. Telemetry and telecommands use the Consultative Committee for Space Data Systems (CCSDS) international standard.

Telecommunications use S-band transmission for the communication with the satellite, and X-band to transfer the scientific telemetry.

The acquisition and safehold mode is used after separation from the launcher and later in case of anomaly detection. The normal mode is the mode in which the scientific mission is carried out (NADIR direction, three axes stabilized by three reaction wheels). The attitude is given by the star sensor. The orbit control mode (MCO) is dedicated to perform the orbit maneuvers. During this mode, the attitude is just provided by the gyrometers. Each burn sequence is preceded by an attitude maneuver, performed using reaction wheels as actuator, to steer the thrusters along the direction of thrust.

Similar to the other MYRIADE satellites, the PARASOL redundancy philosophy is minimized. Spare units exist only for very critical parts such as the solar arrays drive mechanism, onboard transmitter and receiver, and some internal components of the onboard computer.

B. CALIPSO Small Satellite

The CALIPSO is a 635-kg satellite with a power of 560 W based on a PROTEUS platform. It has been successfully launched on 28 April 2006 with a DELTA 2 launcher from Vandenberg, Air Force Base, California, in a double launch, associated with another A-Train satellite, CLOUDSAT.

CALIPSO provides atmosphere vertical profiles measured by a payload composed by an active LIDAR, an infrared imager radiometer, and a visible camera.

The attitude control shall maintain a pointing accuracy better than 0.025 deg. It is performed using magnetometers, gyros, coarse sun sensors, star strackers, and GPS sensors and magnetotorkers, reaction wheels and thrusters as actuators. The actual use of the sensors and of the actuators depends on the satellite modes. The orbit control can use up to four thrusters to perform maneuvers. Electrical power is provided either by the solar cells of the two identical wings of the solar generator, or in case of lack of sun, by the lithium-ion battery. The command control is based on a fully centralized architecture, using a μcs 31750 processor and a mass memory including six stacks of 512 Mbits. As PARASOL, CALIPSO telemetry (TM) and telecommands (TC) use the CCSDS international standard; satellite and mission telecommunications use S-band transmission and dedicated mission transmission use X-band.

As far as possible, the spacecraft is considered as built with two independent half-satellites: one nominal half and one redundant half. However, critical equipment or equipment of more than two units (for instance, gyros, reaction wheels) is shared by both half-satellites. This architecture allows to cope with the satellite performances and the low cost target.

The two CALIPSO main modes are 1) the safe mode, used after separation or in case of satellite failure detection (generally speaking, any satellite failure leads to safe mode, while any instrument current anomaly leads to the instrument passive state, platform remaining nominal), and 2) the nominal mode in which, as PARASOL, the scientific mission is carried out.

The onboard failure detection and recovery function is aimed at setting the satellite in a secure state in case of failure, while being robust to spurious events to preserve the mission availability.

III. PARASOL and CALIPSO operations

Operations can be divided into four parts: 1) low-Earth-orbit phase (LEOP) operations including the launch phase, 2) begin-of-life operations, including activities linked to the ascent phase (maneuvers) as well as assessment tests for the whole satellite including the payload, 3) the mission phase itself, and 4) the end-of-life operations, which can be ended by a reentry phase for the satellite.

All of these phases are prepared before launch, operations are validated, and teams are prepared during training phases and general rehearsals. These training phases begin about one year before launch.

The LEOP phase is a relatively short (about three days) but a very intensive phase in terms of activities to be performed. For PARASOL, this phase was also critical because the satellite was launched with five other satellites by an ARIANE 5 launcher, and that five of them were also operated at CNES. Specific attention

was taken concerning the collision risk between these satellites and the launcher third stage, the consequence being that spacecrafts were not allowed to perform their maneuvers simultaneously for the first days in orbit. For CALIPSO the same criticism exists because the launch is shared with CLOUDSAT.

The mission phase is of course the main part of the satellite on-orbit life, and it consists of repetitive tasks needed to maintain the satellite in nominal conditions to satisfy all requirements in order for the mission to become a success. This phase includes all maneuvers needed for the station keeping, as well as exceptional operations.

A. Operations Linked to the A-Train

PARASOL and CALIPSO are part of the A-Train, which is a constellation of six satellites coordinated by the Constellation Coordination System (CCS at NASA Goddard Space Flight Center). The begin-of-life activities include all maneuvers required to join the constellation (see Fig. 1).

For PARASOL it has been necessary to perform five maneuvers (semimajor axis combined to inclination maneuvers) to join the A-Train, since it had to comply with the main passenger injection orbit: −43 km above AQUA altitude, −0.08 deg for the inclination, and −3,25 deg for the right ascension of ascending node with respect to AQUA plane. This phase lasted about four weeks, and began after the assessment tests were completed, about two weeks after launch. CALIPSO has joined the A-Train in less than one month, performing two maneuvers achieved by the end of May 2006, nearly one month after launch.

Once arrived in the constellation, CALIPSO station keeping is obtained by following a reference grid with maintenance of the ground track at the equator. For PARASOL, the choice was to implement slavery on relative orbital position, station keeping between a target (AQUA) and a chaser (PARASOL). In addition, CLOUDSAT and CALIPSO respect the formation flying rules, with CLOUDSAT slaved to CALIPSO, CLOUDSAT being located within CALIPSO control box limits.

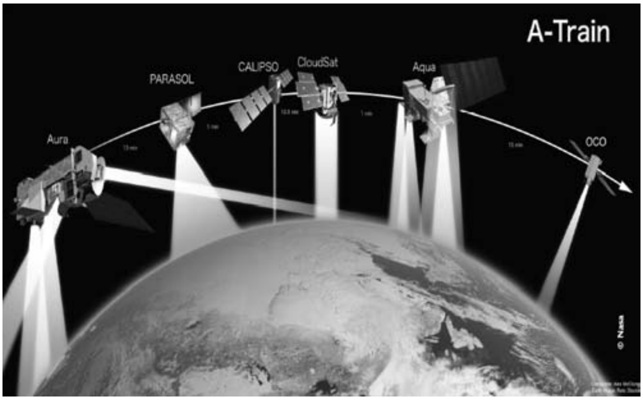


Fig. 1 A-Train constellation.

These choices need a "close" coordination of satellite operations, as well as very good exchanges with the NASA CCS team. This is achieved using daily ephemeris automatic exchanges between AQUA, CALIPSO, CLOUDSAT, and PARASOL satellites operation control centers via a NASA-dedicated server, and monthly teleconferences between satellite operation managers and with the CCS team. Meetings are also planned to coordinate some specific operations concerning the whole constellation. The global surveillance of the A-Train is insured by the CCS at NASA Goddard Space Flight Center in Washington.

When needed, a drag make up (DMU) or a braking maneuver is programmed, so that PARASOL (respectively CALIPSO) stays within its A-Train control box limits. For both satellites, these maneuvers are actually realized every two or three months.

B. Routine Operations

For MYRIADE microsatellites, the operations are based on three basic concepts. The first one is that all ground processes shall be automated and routine operations shall be performed during working hours of working days. The second one is that MYRIADE satellites shall have the sufficient level of autonomy to support several days in orbit without being operated from the ground: a high level of autonomy during acquisition/safe-hold phases, low redundancy level, and few ground interventions during LEOP phases. If an anomaly that could endanger the satellite occurs, it shall be designed to enter in a secure safe-hold mode, and support several days without any ground intervention. Then, if these concepts are respected, MYRIADE satellite activities planning becomes quite simple, and this is the case for PARASOL. Indeed, these rules being respected, ground operations are minimized.

The onboard transmitter is programmed twice a week (switched on before a pass and immediately switched off after the pass, seven days planned), and the TC plans for the satellite attitude control (satellite guidance and solar array mechanism guidance) are also uploaded twice a week (eight days planned). The mission program is uploaded once a week (eight days planned each time). The onboard UTC time is also refreshed once a week. Every day, the orbit determination is automatically computed, the PARASOL location within its A-Train control box is calculated, and the mission scientific products are automatically generated. After each pass, the telemetry is automatically monitored to check the satellite general status and archived on dedicated computers.

Once a month, when the moon enters in the star tracker field of view (two times per orbit for 5–10 days), the satellite attitude control is modified so that it is achieved with the gyrometers instead of the star tracker, which is no longer able to fulfill its requirement. During this time period, programming a station-keeping maneuver (DMU or braking maneuver) is not allowed, even if it is still possible to perform a contingency maneuver if needed.

Monthly, the calibration need is evaluated for some units such as gyros and magnetometers, and the solar sensors status is checked.

For PROTEUS satellites the operation strategy is very similar to the MYRIADE one. The differences are mainly linked to the fact that PROTEUS satellites have more redundant units (implying more operations, more monitoring).

CALIPSO routine activities are also performed automatically. The onboard transmitter is never switched off (then no TC plan linked to this activity), the

satellite guidance and SADM guidance is uploaded every day, and the mission plan is uploaded once a week. The onboard UTC time is also refreshed once a week. The automatic orbit determination, associated to the PARASOL–CALIPSO orbits comparison allows to check that satellite local times respect the scientific requirement (PARASOL shall be maintained 11 min behind CALIPSO in local time).

CALIPSO other occasional activities consist of equipment or subsystem expertises (to check the spacecraft integrity and perform a trend analysis), temporary spare equipment switching on, payload managing during sun eclipse by the moon and randomly solar flares management (consists of switching off the instruments or the whole payload depending on the level of the solar flares), Hubble Space Telescope avoidance (consists of switching off the CALIPSO laser to avoid any Hubble Telescope dazzling), and maneuvers according to other A-Train satellites.

Each day, for each of the two satellites, the A-Train control box is calculated thanks to AQUA ephemeris and dedicated tools, and the position of satellites in their respective control boxes is monitored. For PARASOL, Doppler one-way allows the orbit determination, as for CALIPSO the onboard GPS is used. A station-keeping maneuver is executed about 7–15 days before the box exit. For PARASOL, a specific check is needed to plan the maneuver outside the time periods where the STR could be dazzled by the moon. For CALIPSO, a close coordination with the CLOUDSAT team is needed, because a CALIPSO DMU maneuver implies a CLOUDSAT immediate DMU (executed no later than 24 h after CALIPSO).

Because the ground stations are shared by the PROTEUS and MYRIADE missions, the passes planning is realized once a week, taking into account all on-orbit missions requirements. This is achieved partly by an automatic process, but it also requires modification by hand to take into account all exceptional satellites (i.e., maneuvers) or ground activities (i.e., ground station maintenance). Today, the pass planning, managed by a software called ARAMIS, takes into account three satellites, PARASOL, CALIPSO, and DEMETER (the first MYRIADE satellite, in orbit for two years), and five ground stations.

During the mission lifetime, an operations coordination meeting, involving all the mission team, is held once a week for each mission. During this meeting, all the operations carried out since the last meeting are summarized, the exceptional activities for the week following are planned and coordinated, the Earth terminals reservations are approved on ARAMIS in coordination with other missions, the sequence of events describing all the activities of the next week, as well as the weekly sequence to be uploaded the day after and pertaining to the week following, are approved.

After the operations coordination meeting, all processes are programmed by the operator for the next seven days, thanks to a task scheduler software called "AGENDA." The daily or weekly TC are automatically generated. The telemetry is also automatically stored and is available to the experts on an internal web server. Therefore, all daily and nominal activities are performed thanks to automatic processes, all being under the operator control. If an anomaly occurs, an alarm is generated and the operator is automatically informed by phone. Then, after a first investigation, he calls the adequate expert (ground engineer, flight

dynamics, etc.). At least one pass per day and per satellite is also followed by an operator, who can contact the satellite responsible if an onboard alarm is detected. In any case, the mission operation manager is informed.

IV. Ground Segments

CALIPSO and PARASOL ground segments are both composed of a Mission Operations Ground System (MOGS) and a Satellite Operations Ground System (SOGS).

The CALIPSO MOGS includes two major subsystems: the mission operations control center located at the NASA Langley Research Center in Virginia and the Payload Data Delivery System (X-band station and network).

The PARASOL MOGS is divided in two parts. A first part is called Mission Center Level 1, located in CNES Toulouse Space Center, and which is dedicated to the nominal payload management (checking the payload status, programming the payload, getting the raw mission and process the telemetry to get the level 0 and 1 products, delivering and archiving the data). A second center, called “ICARE,” is located in the city of Lille, where one aim is to pool data and provide shared services enabling the scientific community to exploit the huge volumes of data derived from the A-Train.

Both PARASOL and CALIPSO SOGS are based in CNES Toulouse Space Center. This part of the ground segment is used to operate the satellites, perform the monitoring and platform controls, orbit and attitude control, payload service, and satellite expertise.

CALIPSO and PARASOL SOGS are very similar and use the following components (see Fig. 2):

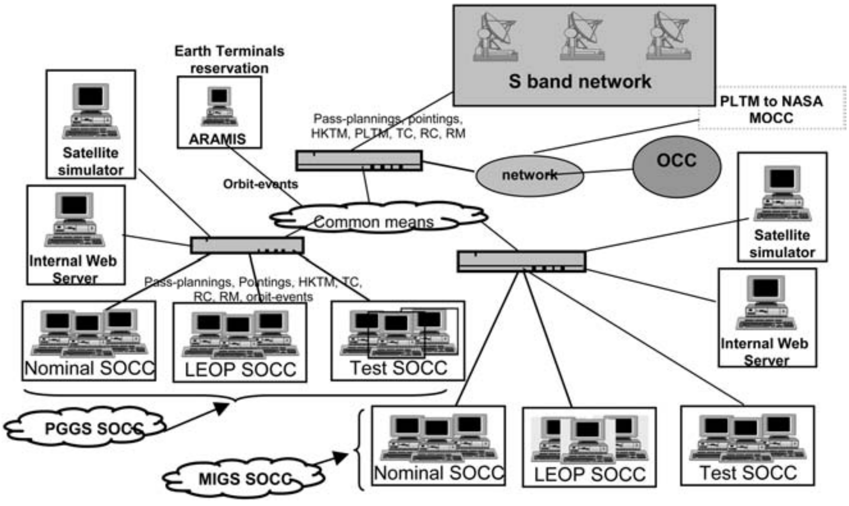


Fig. 2 CALIPSO and PARASOL SOGS.

1) A common network of S-band Earth terminals called (telemetry and telecommand earth terminal) (TTCET) located at Kiruna, Sweden, and Aussaguel, France. They are all based on the same architecture, ensure the TM/TC link with the satellites, and can download Doppler measurements for the orbit control.

2) One or more CNES 2-GHz stations that can be “kitted” to be adapted to the small and microsatellites ground constraints (e.g., one station located in Hartebeesthoek, South Africa, another one in Kiruna).

3) An X-band station called TETX, which is dedicated to the acquisition of high-rate mission telemetry. This station is located in CNES Toulouse Space Center and is only used by the MYRIADE satellite product line.

4) A common mean called “ARAMIS” used to plan passes for all in-orbit PROTEUS and MYRIADE satellites.

5) A common archiving system.

6) A common configuration management system.

7) Two separate satellite operation control centers (SOCC). These SOCC are designed to support up to seven on-orbit satellites belonging to five different missions. Because of the actual difference between PROTEUS and MYRIADE satellites, these SOCC are not shared, even if they are very similar. PROTEUS and MYRIADE SOCC have the same architecture, based on the use of personal computers. The main functions are TM/TC real time management, telemetry display using mimics, orbit and guidance management, telemetry monitoring and archiving, and file transfer management. In a general manner, ground software programs are shared between PROTEUS and MYRIADE, but their configurations are different, satellite database management is specific to each family. Moreover, the mission interface management is specific to each mission.

8) A common data transmission network (RTD) that provides the connection between the stations and the CCC, the stations and the missions centers, the SOCC and the mission centers. The network architecture relies on the multimissions resources at CNES.

For LEOP phases, as well as for commissioning, this system can be completed by Earth terminals of the 2-GHz network, which provide additional angular measurements used by the orbit control center for the orbit determination.

There are three SOCC per satellite family (PROTEUS and MYRIADE), one used for the routine operations for all the satellites of a same product line (with respect to the limitation of seven satellites or five missions per SOCC), one used for qualification and LEOP phases, and the last one dedicated to testing phases, with satellites simulators.

V. Operation Management and Associated Manpower

The operation team for each mission is composed of a satellite-responsible group, a ground-responsible group, a flight dynamics engineer, the mission programming specialist (if needed), and an operator for the routine monitoring, this team being managed by the mission operation manager. When needed, the team can be completed by satellite experts (e.g., thermal control, power, orbit and attitude, software, command and control experts) and ground system experts (e.g., network, ground station experts).

Ground experts can intervene independently on PROTEUS or MYRIADE ground segments, since both are very similar. Moreover, to respect this similarity, one engineer is responsible for the whole coherency and the operational management of both ground segments (SOGS).

Flight dynamics engineers are also able to intervene on both CALIPSO and PARASOL missions, as well as operators. As for satellite responsibility, there are two teams, one dedicated to microsatellites and the other to small satellites. Indeed, PROTEUS and MYRIADE platforms are too different to be compatible with a shared team of satellite experts.

This organization allows working with a minimum manpower available only during working hours of working days. During weekends and days off, processes are entirely automatic. Moreover, the telemetry is automatically monitored after each pass, and if some predefined critical anomalies are detected, a dedicated software called SYGALE automatically phones a 24/7 manpower person and delivers the appropriate message to this person, so that he can intervene on the next pass. Some actions may be needed to put the satellite in a secure safe-hold mode. A first investigation is made to prepare the activities of the next working day. This is only possible with thanks to these specific platforms that are designed to withstand a safe hold mode of several days without damages.

When CALIPSO and PARASOL are in nominal mode, the routine operations, as previously described here, can be foreseen and performed automatically. During nominal activities, spacecraft status is checked automatically by monitoring telemetry, and in the case of noncompliance, alarms are raised. These alarms are shared in two categories: yellow and red alarms. If yellow alarms lead only to expertise during working hours, some specific red alarms may require a rapid intervention from the ground, even during days off.

During working hours of working days, should any red anomaly be encountered (ground or onboard anomaly), a hotline is automatically called. In this case, an expertise shall be performed and actions are undertaken to recover a nominal functioning.

The most important red alarm corresponds to the safe mode (transition performed automatically by the onboard software in case of main failure detection). The return to nominal mode is performed according to a dedicated sequence, shorter on MYRIADE satellites as PARASOL (two days) than on PROTEUS satellites as CALIPSO (five days).

During nights, weekends, and days off, MYRIADE and PROTEUS families have different levels of intervention. Indeed, the only sequence for which a rapid intervention is needed for MYRIADE microsatellites is the detection of a safe-hold mode. In this case, some telecommands shall be sent to the satellite to protect the battery capacitance. Then, if a reset of the onboard software is detected during the TM monitoring (done automatically after each pass), an on-call 24/7 satellite-responsible person is contacted by the hotline. This person shall intervene on the next pass to send the correct TC. If the ground processes are blocked, a ground-responsible person is also contacted to secure the operation. On PROTEUS satellites, other red anomalies could need a rapid intervention of satellite responsible, typically those which could impact the availability of scientific products (in addition to those which can lead to the safe mode).

Other extra activities are always performed with manpower, for obvious safety reasons. Depending on the activity itself, they can be performed either during working hours or non-working hours (nights, weekends, and days off).

This is the case for the launch phase, the begin-of-life activities, and the rise-up maneuvers that can be undertaken nights and days. For PARASOL and CALIPSO, because of the location of the ground stations, all passes are in the range of 10:00 p.m. to 2:00 a.m. (no pass between 2:00 a.m. to 10:00 p.m.). This explains why the beginning-of-life activities require bigger teams since the working hours are not sufficient to follow all passes.

Other unusual activities (e.g., onboard computer new software upload, emergency maneuver) are planned, as possible, during working hours, and with a reenforced team.

VI. Experience Feedback

Since June 2004, when DEMETER, the first microsatellite from the MYRIADE product line, was launched, until today, six MYRIADE microsatellites and one PROTEUS small satellite, CALIPSO, have been operated from CNES Toulouse Space Center (JASON, the first PROTEUS satellite, is operated in partnership with NASA).

Operations of these seven satellites are reduced to a minimum, thanks to autonomous platforms and ground automatic processes. For each satellite, the number of passes have been reduced to a minimum, for example, four passes per day above S-band ground stations for PARASOL, that is 25 min of visibility for the platform telemetry and telecommand, and four passes per day above X-band ground stations, that is 20 min of visibility, to download the scientific data.

A lot of lessons have been learned these last years, on several domains as well as onboard operations, ground activities, organization, and management.

For the MYRIADE product line, the safe mode robustness has been tested several times in orbit. Nevertheless, this way of functioning has an interest only if the operations needed to recover the nominal operating mode are reduced. For MYRIADE satellites, these operations can be managed in two days (with fewer than 10 passes).

The DEMETER on-orbit experience feedback has allowed uploading corrections on the PARASOL platform software very soon after its launch. On the contrary, anomalies that appeared first on PARASOL have been corrected on DEMETER very rapidly. This was also made easier because the team was shared between these microsatellites.

Operations preparation is less important on these kind of platforms since redundancy philosophy is reduced. This is not the case for the functional “end-to-end” tests that are under the satellite project team responsibility.

Concerning the ground segment, the automation of daily sequences is a great success. However, the major difficulty was to obtain an actual routine mode, with fully operating automatic processes and minimum manpower. Indeed, if ground tests before launch are sufficient to assess that processes are secured, they are not close enough to the actual daily life where unexpected external noise (such as network problems) can easily disturb the whole system and, by the way, stop the process.

It has taken a long time to come to a stable mode, and the PARASOL first six months in orbit required significant (and in fact non-anticipated) manpower. Moreover, the ground system is complex, with a great number of interfaces, and it needs to have a permanent capability of adaptability, development, and expertise. Even if the system is fully automatic, this functioning requires having operators during working hours, for the global monitoring, and also needs permanent updating of software and computers.

One more difficulty was the implementation of an automatic process for the passes reservation. Today ARAMIS is not fully capable of insuring compliance with all mission needs. Then, if the first loop of reservation is automatic, a manual verification is unavoidable.

For the management of the whole activity, the operation coordination meeting held once a week is in accordance with the need. This meeting is also held each time an unplanned and urgent activity has to be discussed (for example if a safe hold mode occurs). Moreover, if all topics cannot be discussed during this meeting, which lasts about 1 hour, then other meetings have been created to coordinate activities and priorities between the different missions, should they be under preparation or already in orbit.

One major challenge is to have relatively short preparation phases, and to enter rapidly into a routine mode with a fully performing system, since micro- and minisatellites in-orbit lifetime is short (from one to three years, compared to telecommunication platforms). Then all beginning-of-life onboard or ground anomalies have to be solved as soon as possible. This is facilitated by the organization itself, since the satellite project team is shared between new satellites under development and satellite under exploitation, and since the operational team is involved in the very beginning of the project.

VIII. Conclusion

If 2004 was the year of MYRIADE satellite launches (six satellites launched and operated at CNES Toulouse), then 2006 was the year of PROTEUS satellites: indeed, CALIPSO launched in April and COROT mission nine months later will increase the operational activity. From 2007 to 2010, several other satellites should join the pool (SMOS and JASON2 satellites belonging to the PROTEUS family and PICARD, MICROSCOPE, the four ELISA of MYRIADE satellite family).

During this period, some processes need to be reviewed and to improve the automation of tasks. In particular, the passes reservation loop, achieved with the ARAMIS software, will have to take into account in the near future several other satellites, such as COROT, SMOS, PICARD, and MICROSCOPE.

Another important work concerns the unification of the different CNES networks used for data transmission between ground stations, SOCC, and MOCC for satellites operated by CNES. The major advantage is the improvement of the network supervision for MYRIADE and PROTEUS satellites, especially during weekend and days off. Until now, this task is not programmed for these two families, and its implementation will increase the data availability.

Today, the MYRIADE and PROTEUS satellites operation management (automation of ground processes, project team, and operation team organization) is a great success. Indeed, after some problems essentially linked to the youth of

satellite families and operating systems, the scientific data availability is now greater than 95% for on-orbit satellites. Scientific products are available to the scientific community the day after their generation in orbit. This shows that this kind of operation management of low-cost satellite families can be fully compliant with scientific requirements.

References

[1] Dubernet, N., "Operations Concepts for Orbit Control of Micro-Satellite Scientific."

[2] Gelie, P., and Crebassol, P., "CNES PROTEUS/MYRIADE: Two Low Cost Small Satellites Families," *Proceedings of the 5th International Symposium on Reducing the Cost of Space Ground System Operations*, Pasadena, CA, 2003, http://descanso.jpl.nasa.gov/RCSGSO/Proceedings/proceedings.html.

Chapter 25

Furthering Exploration—International Space Station Experience

Gary Kitmacher* and Theresa G. Maxwell†

NASA Headquarters, Washington, D.C.

I. Introduction—The Vision for Space Exploration

ON 14 January 2004, President George W. Bush announced the Vision for Space Exploration (VSE). It establishes a course that expands the human presence beyond the Earth—first, in near Earth orbit on the International Space Station (ISS); then in the next decade, to the moon; and later, to Mars and beyond. NASA has unveiled plans for the next generation spacecraft, the Crew Exploration Vehicle (CEV), which will take us there.

Completing assembly of the ISS by the end of the decade, and fulfilling commitments to the international partners, is a crucial first step in human exploration. NASA is refocusing ISS research to meet the VSE requirements. As humans venture further from Earth, and as program timetables and mission logistics increase in time, distance, and complexity, it will be crucial to have crews and vehicles that can be sustained with greater reliability in the harsh rigors of space. The ISS mission can directly support these agency needs in the following areas:

1) Develop, test, and evaluate biomedical protocols to ensure human health and performance on long-duration space missions.

2) Develop, test, and evaluate systems to ensure readiness for long-duration space missions.

3) Develop, demonstrate, and validate operational practices and procedures for long-duration space missions.

II. International Space Station Experience

The ISS is a technological undertaking of global scope. Elements of the ISS are provided and operated by an international partnership of governments and their

*Senior Engineer/Manager, International Space Station Program, NASA Johnson Space Center. Senior AIAA Member.

†Senior Engineer, International Space Station Division, NASA Headquarters.

contractors. The principals are the space agencies of the United States, Russia, Europe, Japan, and Canada.

The ISS has been continuously crewed for more than five years and is about 50% complete with approximately 186 metric tons of mass on orbit. There are 15 elements in orbit today, 9 elements ready for launch at the Kennedy Space Center in Florida, and 7 elements in process at international partner sites. When assembly is complete, the ISS will comprise 453 metric tons (nearly a million pounds) of hardware, orbited in about 40 separate launch packages over the course of more than a decade. To date, there have been over 50 flights to the ISS, including flights for assembly, crew turnaround, and logistical support.

Figure 1 depicts the final configuration of the ISS when assembly is complete and identifies those elements that are currently on orbit and those that are awaiting launch.

NASA will use the space shuttle, prior to its retirement in 2010, to complete the ISS assembly. Assembly priorities are to 1) complete the truss segments; 2) establish the life support, thermal control, and power systems that can sustain the assembly-complete station; 3) attach the international partner elements, including the Japanese Experiment Module (JEM), the European Columbus Module, and the Canadian Dextre robotic manipulator; and 4) provide the logistics to sustain the ISS.

Russia will launch its remaining assembly elements to the ISS, including the Multipurpose Laboratory Module and the Research Module.

The final ISS configuration will support growth to six crew members in 2009 with the delivery of additional crew quarters, galley, waste management system, and new oxygen generation system. During this period, the Russian Progress vehicle will be used to augment space shuttle logistics capacity, and the Russian Soyuz vehicle will be used for some crew rotations. Once operational, the

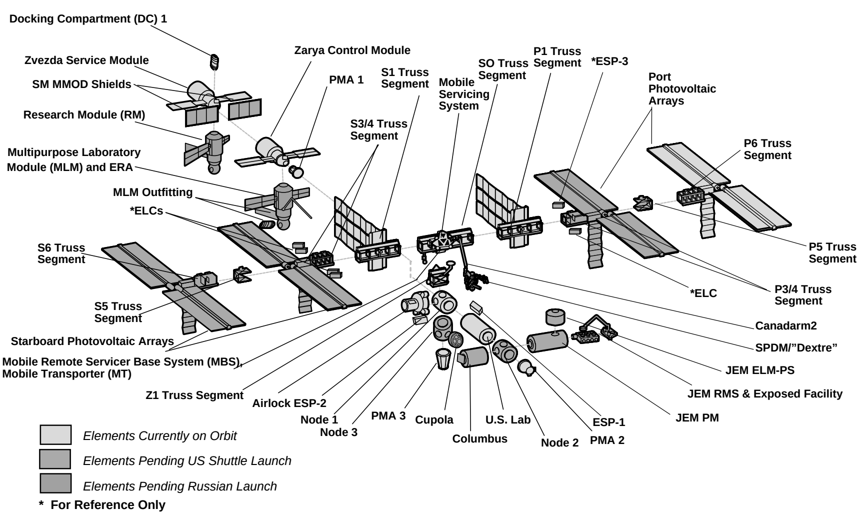


Fig. 1 International Space Station configuration at assembly complete.

European Automated Transfer Vehicle (ATV) also will be used to supply logistics.

Once the shuttle is retired in late 2010, NASA and its international partners will use a combination of their collective assets to support and maintain the ISS in orbit. The Russian Soyuz can be used to carry crew while the Russian Progress, European ATV, and Japanese H-II Transport Vehicle (HTV) can be used to share the burden of logistics support. NASA is also seeking a commercial provider to supply logistics and crew transport to the ISS. Within two to four years after the space shuttle's last flight, the new NASA Crew Exploration Vehicle should be ready to support flights to and from the ISS.

The International Space Station Program has endured now for 22 years—a full generation of technical expertise that has designed, developed, operated, and managed the program. ISS personnel have successfully adapted to changing circumstances, whether driven by technical or operational difficulties, transportation shortfalls, budgetary considerations, or political redirections. This has included several major redesigns of the ISS vehicle, as well as major changes in ISS operations. For example, reductions in launch and return capability after the *Columbia* accident have taught ISS engineers and scientists how to deal with logistics shortfalls and to adapt ISS research to new operational realities.

The Exploration Program will not likely be completed within the careers of the personnel now establishing it. The personnel who will implement the Mars landing may not yet have begun their careers. Through the ISS Program, personnel are developing the experience, knowledge, and skills to overcome the inevitable contingencies that will arise in the Exploration Program. These valuable lessons should be factored into the Exploration Program from the beginning, so that they do not have to be relearned by the next generation of engineers and scientists who will take humans further into the solar system.

As we expand human presence beyond the Earth, first in orbit, in the next decade to the moon, and later to Mars and beyond, the ISS experience can help to guide our success in exploration.

III. Areas of Applicability to the Exploration Program

Through the ISS program, NASA and its partners have acquired experience in building and operating complex space vehicles. The ISS has been a tremendous challenge of integrating the hardware, computer software, command and control interfaces, crew procedures, logistics, ground support teams, and research, with the added dimension of dealing with different languages and cultural paradigms—in the largest, most complex spacecraft ever devised. This technical challenge is certainly one of the most difficult any international partnership has ever faced.

Perhaps as significant as the technological sophistication is the complexity of the multinational and multi-organizational elements involved. The ISS has been the most politically complex space exploration program ever undertaken. It involves multiple aerospace corporations and nearly every international space agency working as program partners. Further, it integrates international flight crews, multiple launch vehicles, globally distributed launch/operations/training/engineering and development facilities, communications networks, and the international scientific research community. Elements launched from different countries and continents

have never been mated together until they reach orbit, and some elements launched later in the assembly sequence had not been built when the first elements were placed in orbit.

The ISS program's greatest accomplishment is as much a human achievement as it is a technological one—how best to plan, coordinate, and monitor the varied activities of the program's many organizations. Getting all of the personnel elements to effectively work together has been a continuing challenge for the program management, regardless of whether they were from the United States or other nations, the various NASA centers, or civil service and industry. The various communities often have differing priorities and are competing for the same resources. The program has succeeded by developing management processes that address the needs and constraints of the various organizational elements. Roles, responsibilities, authority, and interfaces were negotiated and documented. Control boards, reviews, documentation, procedures, and information systems have been designed to facilitate program management and coordination. These ISS operations management processes and tools have continually evolved to accommodate changing needs, to address problem areas, and to take advantage of potential efficiencies. Examples are given throughout the chapter.

The ISS program provides valuable lessons for current and future engineers and managers. ISS provides real-world examples of what works and what does not work in space, as well as lessons in the management of space programs here on Earth.

Specific operational areas in which the ISS experience can be applied to the VSE include crew operations, spacecraft systems operations, and crew–system interface operations.

A. Crew Operations

High-performing crews are critical to successful long-duration missions. Mission failures can result from degradation of human performance, either physiologically or psychologically, after long-duration exposure to the space environment and to the stress of isolation. Specialized skills and training of international crew members, as well as advanced protocols, procedures, and tools, were developed for the ISS and can be used to reduce the risks to future exploration missions.

The interaction of the international crew with multiple mission control centers is also a significant element that can make a space mission highly successful or bring work to a standstill. The ISS provides an environment to improve the interaction between crew and ground and make missions safer and more effective. Working for months with crew members from other countries and cultures is an important aspect of the ISS program. Developing methods to work with our partners on the ground and in space is critical to providing innovative solutions to operations challenges.

1. *Long-Duration Crew Operations*

By necessity, the ISS program adopted crew operations philosophies and support tools that are conducive to long-duration operations.

Unlike rigidly scheduled short-duration missions, long-duration crew schedules must be carefully balanced to provide dedicated work time in addition to crew member time for exercise, hygiene, rest and sleep, and personal time. The number of tasks required to be performed according to a predefined schedule has been minimized; the crew has the flexibility to execute other routine, noncritical, and nonhazardous tasks from a predefined task list. The increased scheduling flexibility permits the crew members to better manage their own activities and time. One crew member commented recently that “the difference between the work week and his weekend is that on weekends he gets to choose the work he wants to do.” This makes work on the space station more Earth-like, providing the crew more autonomy. With greater autonomy the crew member realizes a heightened sense of professionalism, and greater enjoyment and an enhanced feeling of accomplishment. It has also frequently been beneficial in terms of the quantity of work performed.

Some have suggested that for future missions, the ideal of crew autonomy be taken to new levels—use the professional education, expertise, and initiative of the highly trained and motivated crew members as the basis for planning the program of research to be conducted rather than using the crew member as a simple equipment operator.

A new freedom in communications has been realized on the ISS. E-mail enables easy, frequent, and routine communications with professionals, colleagues, friends, and family. Use of the Internet protocol (IP) telephone enables verbal communications as easily as if the crew member is in the office. Routine communications between the crew in orbit and managers or researchers on the ground have expedited the exchange of thoughts and information.

Planned daily conferences at the start of each workday permit flight and ground crews to identify and prioritize tasks requiring attention, and at the end of each workday, to identify the tasks that have been completed. This allows the ground to track mission accomplishments and issues while keeping unnecessary communications to a minimum.

Because current hardcopy versions of crew procedures and flight plans cannot be maintained onboard, the ISS program implemented software systems to electronically view and manage this information. Any needed updates to the procedures are made on the ground then uploaded to the ISS for immediate access by the crew. The onboard crew also has an electronic version of the ISS flight plan. Capabilities are provided for the crew to make annotations on their planned activities and to perform some limited plan editing. Updates to the crew flight plan are uploaded on a daily basis.

Psychological support of the crew member has gained a new level of attention on ISS with proactive review of operations processes and requirements by psychologists and managers on the ground and with routine “care packages” provided from the crew member’s home and family to orbit. In addition to regular communications sessions between crew members and the families, special communications sessions have been arranged between crew members and recognized world experts in science, technology, philosophy, music, and entertainment.

The extent of crew-controlled task scheduling, the degree of crew autonomy, and the importance of psychological support will become more critical as missions become longer, as missions take place at greater distances, and as the potential for

any interruption in communications grows. The Exploration crews can build on this ISS experience. The ISS has been a cornerstone in advancing knowledge about how to live and work in space for long, continuous periods of time, and the knowledge gained will remain critical to our future exploratory journeys.

2. *Crew Training*

The ISS crew must be able to handle both nominal and off-nominal operations. This requires general training on the onboard hardware and systems as well as specific training on the procedures to be performed for specific operations. Effective training is essential because the crews may be required to control or to restore systems in the event of automated systems failures, loss of communications with the ground controllers, or other malfunctions and emergencies. Some ISS systems and crew procedures are quite complex, which can make it difficult for the crew to deal with contingencies if not adequately trained.

The international nature of the ISS led to the development of a training program that is geographically distributed. Each partner is responsible for training the crew on the operations of their respective elements and systems. There are, therefore, training facilities in the United States, Russia, Europe, Japan, and Canada. Scientific equipment training further widens the geographically distributed nature of the training requirements. This adds overhead and logistical complexity to the training schedule, which, if not effectively managed, can result in crew fatigue prior to launch. A two-year or longer training regimen has been required by most long-duration crews. New processes for managing the crew members' time before and after missions have been developed.

The ISS experience has shown that U.S. and Russian training methods for flight crew and ground personnel differ considerably. U.S. training focuses more on specific task and procedural training, while Russian training focuses on overall understanding of design functionality and system operations. Advantages can be seen in both approaches. Generic training on system design and functionality provides a knowledge base that the crew can use when dealing with unforeseen events, while specific task training is beneficial for very complex or hazardous operations.

For long-duration missions such as the ISS, "refresher" training and systems and hardware reference data are especially important in preparing for complex, intricate, critical, or hazardous operations, since the crew's initial training on the operation may have been months or even years earlier, and on the ground. The Crew On-Orbit Support System, developed initially for use on Mir, and expanded upon greatly for the ISS, provides onboard computer based training (CBT) capability for the flight crew. A library of software provides lessons and reference data covering many systems and critical operations and is available for crew use and review. New and impromptu operational procedures have been developed and uplinked for use in space. The ability to effectively train onboard will be key to future exploration missions when Earth-based training last occurred months or years earlier.

Significant investments were made in ISS training resources, processes, and facilities due to the complexity of the spacecraft systems and mission requirements. These investments can now be applied to exploration.

3. *Extravehicular Activity Operations*

To date, there have been 28 space-shuttle-based and 36 space-station-based extravehicular activity (EVA) operations at the space station, totaling over 385 hours. More EVAs are being planned as the assembly continues. The majority of these EVAs have been for assembly tasks, but several were for maintenance, repairs, and science. These tasks were conducted from three different airlocks—the shuttle airlock, ISS Joint Airlock, and the Russian Pirs—using two different space suit designs, the U.S. Extravehicular Mobility Unit (EMU) and the Russian Orlan.

In some instances, collaboration between the U.S. and Russian EVA flight control teams has been particularly close. Control moment gyroscopes (CMG) are used to maneuver the ISS, using naturally replenished electrical power to operate the motion control system, instead of attitude control thrusters using fuel that must be re-supplied from Earth. The system includes four CMGs, even though only three are required for full operation and two CMGs can provide adequate although degraded control. When CMG 1 failed, the cause of the failure of its rotational bearing could not be resolved through telemetry transmitted to the ground. The CMG would have to be replaced and the failed unit returned on the next available shuttle so that the failure could be analyzed. The remainder of the CMG system could continue to maintain vehicle attitude control. However, when a second control moment gyro (CMG 2) lost power because of the failure of its remote power controller module (RPCM), planning for an EVA to change out the RPCM started immediately.

The RPCM change-out EVA was a great example of multinational cooperation, as this was the first EVA to be performed in Russian suits on the U.S. segment of ISS. The Russian flight control team was in control of the EVA while the crew was on the Russian portion of the station. The Russian and U.S. flight control teams worked together flawlessly to assist the crew in translating from the Russian to U.S. segment and back, and in monitoring crew health and Orlan EVA suit status. During the EVA, the Canadian Space Agency (CSA)–developed Space Station Remote Manipulator System was used to monitor the status as the astronauts worked outside the spacecraft.

One major lesson the ISS program learned is the importance of designing and certifying EVA equipment for longer lifetimes, with the capability to perform maintenance in space and with an understanding of the on-orbit certification criteria prior to continued use. The EMUs are normally planned to be returned to Earth on the shuttle for servicing. During the shuttle downtime after the *Columbia* accident, two of the three EMU suits on orbit, as well as the U.S. Joint Airlock, experienced technical issues that prevented their use in support of spacewalks. Root cause of the loss was contaminants in the suit and airlock coolant water that blocked filters and disrupted magnetic coupling of the suit pump rotor. Water pump rotors also had de-bonded over time. The Russian Airlock and Orlan suits were relied upon to conduct ISS EVAs during this period. However, through the ingenuity of the engineers on the ground, and the skills of the crew in space, the EMUs were repaired and made serviceable. Although these EMUs were not used for an EVA, this was a breakthrough in the normal maintenance philosophy for the EMUs, as all critical maintenance had previously been performed on the ground. The ingenuity of the ground team and the crew members was demonstrated by

developing the procedures to troubleshoot and repair the EMU cooling pump impellers on orbit with no training and limited tool selection. U.S. EVA capability on the ISS was not fully restored until the July 2005 shuttle flight, which replaced the two EMUs and delivered a filter/iodinization kit that was successfully used to complete airlock restoration, and will continue to be used in the future to assure EVA readiness. The lesson for the exploration program is to design its EVA equipment for in-situ servicing and repair and to develop specific criteria for continuing certification.

The ISS experience with EVA training is applicable to other long-duration missions, such as a journey to Mars. The space shuttle EVA training philosophy has been to train crew members on the specific tasks to be accomplished, in the specific order they would be performed. A space-shuttle-based EVA is a well-practiced and carefully orchestrated ballet, in which everyone knows his or her part by rote. ISS crews, on the other hand, may be faced with both planned and unplanned, or contingency, EVA tasks. Time and resources in which to prepare the ISS expedition crews for EVAs are limited. To most efficiently use the available preflight crew time and training resources, a different philosophy has evolved based on crew member recommendations. This new philosophy is to train the crew members on a skill set that is applicable to most EVA tasks they will encounter. If there is an especially complex task required of a crew, some specific task-based training may still be required. This skills-based philosophy prepares expedition crew members to be able to react to nearly any EVA contingency or repair task that might arise while they are on orbit. This philosophy has repeatedly shown its value during several unplanned EVA tasks that were required to replace failed external hardware on the ISS.

Preflight training on both the U.S. and Russian EVA systems is augmented with on-orbit training. Each EVA is preceded by an on-orbit training session, in which the EVA crew members review their procedures and practice the EVA, including donning/doffing of the suits. These sessions can be used to train the crew on-the-fly for EVA tasks that were not planned preflight.

Another aspect of EVA operations that should be considered when designing exploration missions is the extent of EVA preparation and support that is required. Each EVA requires a significant amount of crew time in addition to the actual EVA. Besides the preflight and on-orbit training requirements, numerous operations must occur immediately before and after an EVA, including preparing the airlock, inspecting the suits, pre-breathe protocol procedures, servicing the suit after an EVA, and closing out the airlock. This additional overhead should be considered when defining EVA requirements and strategies for the Exploration Program.

During nominally planned ISS EVA operations, the EVA crew is supported by one or more crew members inside the vehicle. When three crew members were available on the ISS, a crew member inside the ISS supported the two EVA crew members outside. When the ISS crew was reduced from three to two crew members in the wake of the *Columbia* accident, EVAs became two-person operations. With no one remaining inside the vehicle, systems monitoring and spacecraft operations are turned over to mission control. The ground also assumes the role of the third crew member in helping to coordinate the EVA. This kind of operation is not new to either the United States or Russia. During the Apollo moon landings

the crew worked on the moon's surface while ground controllers monitored the spacecraft systems. During Salyut and Mir, Russian cosmonauts routinely left the spacecraft untended during spacewalks. This mode of operation is possible as long as the ground has the ability to monitor and control the vehicle.

The operational lessons of the ISS in the areas of EVA suit maintainability, training, and EVA support may prove critical for long-duration crewed missions that venture even further from the Earth.

B. Spacecraft Systems Operations

Efficient, reliable spacecraft systems are critical to reducing crew and mission risks. Optimizing systems performance and characterizing system performance in space will reduce mission risks and advance capabilities for planetary distances and autonomous vehicle and systems management.

Confidence in life support systems for water and waste recovery, oxygen generation, and environmental monitoring technologies becomes more critical as the distance and time away from Earth increase. The ISS is the first space vehicle in the U.S. space program in which reliance upon recycled water and oxygen has been critical to continuation of the mission. For the first time, NASA engineers have developed the closed-loop recycling systems and tested them in space. The ISS is NASA's closed-loop life support test bed for demonstrating these advanced capabilities. The systems will serve as the basis of the expertise required to send crewed spacecraft to the planets.

Maintaining crew health is key for long-duration flights. The ISS must provide exercise and environmental monitoring systems that are in use continuously over many years of operation. ISS also proved an important lesson in defining the criticality of these systems. Much has been learned about developing exercise equipment and its effectiveness for maintaining crew fitness in microgravity. More long-duration experience with these systems is needed before extended missions on the moon or to Mars are attempted.

Operations protocols and support tools that minimize the ground support infrastructure needed to monitor and control spacecraft systems are also essential for long-duration missions. The ISS operations concepts and ground facilities continue to evolve due to ongoing efforts to increase effectiveness and minimize operations costs.

1. System Design for Long-Term Operations

The United States and Russia evolved different approaches to system design and operations. The ISS experience has shown that, for long-term operations, there are advantages and disadvantages to both approaches.

The Russian modules and systems of ISS are essentially identical to those used in the Russian Mir station and were developed beginning with the Salyut designs of the early 1970s. Russian design philosophy embraces simplicity and robustness. Many of the systems, however, require frequent crew interaction for maintenance and operation. The systems are usually reliable and easy to operate and, when maintenance is required, permit crew access and interaction. Emphasis is placed on operability and functionality, but the minimal telemetry means that systems

may unexpectedly malfunction before corrective measures can be planned. The on-orbit crew is expected to operate with a level of independence from the ground that requires the crew to take on the responsibility to ensure the systems remain operational. Russian system reliability is based on periodic maintenance and component replacement.

Most of the U.S. modules and systems now part of ISS have little heritage from prior space flight programs. The U.S. systems tend to be more complicated than their Russian counterparts. The U.S. systems provide considerable data to flight controllers via telemetry. This allows the crew to rely on the flight control team to monitor the performance of the systems. Frequently ground controllers have more data than the onboard crew, and they may have more control than the onboard crew. Most of the U.S. systems are computer controlled. This permits a high degree of automation and ground monitoring and control, but this also means the systems may not operate at all unless computers and software are operating nominally.

ISS has shown that a system driven by computers and software must have sufficient redundancy; the design must accommodate the microgravity and radiation environments, and a priority factor to consider in the software design from the standpoint of operations is the ease of crew interaction and control. The ISS experience has shown that crew input into the design of onboard computer displays is essential to ensure that the displays are intuitive, easy to navigate, and contain the information needed for effective crew operations, especially in a time-critical situation.

The U.S. command and data handling (C&DH) architecture is a tiered approach with the triple redundant command and control (C&C) computers providing the top level of control. Prior to the installation of the U.S. laboratory, the U.S. node was controlled with two of its own computers and software. Once the laboratory was integrated to the ISS and activated, the C&C computers in the laboratory took over the top level of control of the U.S. C&DH architecture. The node computers continued to control the node functions and communicated with the C&Cs. The node computers also retained a safety net function referred to as Mighty Mouse. In the event that all three C&C computers should fail, the Mighty Mouse software would activate and the node computers would take over a limited portion of the C&C's role and attempt to restore the C&C computers. Approximately four months after the laboratory activation, all three C&C computers did fail within hours of each other. The Mighty Mouse function performed as designed and kept the critical U.S. systems running. Failure analysis revealed that the cause of the C&C failure was related to mechanical issues with the hard drives on those computers. The hard drives were replaced with solid state memory units (SSMU) and there have not been any problems since. The potential failure of the hard drives had been identified some time before, and development of the SSMUs had begun nearly a year before the on-orbit failure, allowing replacement less than nine months from the time of the failure.

Laptop computers are used both as the crew interface to the C&DH and to perform less critical functions such as display of procedures, e-mail, and IP phone. Laptop hardware has been upgraded twice since First Element Launch, providing a higher performance platform than would have been possible with more traditional avionics equipment.

The maintenance of avionics software on ISS has been another success story. The software upgrade process was originally launch driven, with most initial software loads resident on the computers launched as part of each ISS element. Software updates (both patches and new releases) were provided via uplink or through the use of CD ROMs flown on the shuttle, Soyuz, and Progress. After the *Columbia* accident, the update process was continued, and virtually all of the space station's U.S. and Russian software has been upgraded at least once since. Continuing the software update process in the absence of shuttle flights has allowed the implementation of software fixes and improvements with a corresponding reduction in operational workarounds and increase in operational efficiencies. In addition, this approach has allowed the software team to live within decreasing budget allocations planned before the *Columbia* accident.

2. *Habitation and Life Support*

The ISS is demonstrating the importance of habitability in sustaining crews and spacecraft operations over the long time periods that will be critical for lunar and planetary habitats and Mars transit vehicles. Habitability issues are important for maintaining crew health and feelings of well-being. Inadequate attention to habitability presents serious mission and safety risks.

Noise levels were a concern from the outset of the ISS Program, beginning with requirements definition. Inadequate attention in the design and development stages and, in some cases, use of decades-old technologies, combined with inadequate noise standards, led to a noisy environment in which personal hearing protection for the crew has become the norm. In some circumstances, the noisy environment makes it difficult for crews to communicate with one another or with the ground. As systems are replaced, this noise problem is being reduced, but on long-distance missions upgrades are not an option.

Reliable operation of the life support systems in human spacecraft is critical and will become much more significant as crews and spacecraft venture further from their logistics source on Earth. Dissimilar redundancy in key life support systems has proven critical to the ISS. The United States and Russia used different hardware design reliability philosophies. The Russian systems are made of modular, stand-alone hardware. Though these components endure periodic failures and anomalies that reduce performance, frequent, simple maintenance can keep the systems operating, and when there are more significant problems, replacement components or assemblies can be launched on Progress logistics missions. The U.S. systems were designed independently from the Russian systems, are more complex, experienced different operational failure modes, and required varied maintenance and repair solutions.

The Russian "Elektron" system, for example, has been the primary generator of oxygen onboard ISS. Its major component, the "liquid unit," generates breathable oxygen by electrolysis of water recovered from the cabin air and separation into oxygen and hydrogen. A series of failures of the fluid micropumps, caused by air bubbles and contaminants in the fluid lines, occurred during the shuttle down period. The failures necessitated the change-out of three liquid units in succession, and then considerable hands-on maintenance by the crew members to maintain partial operability. Replacing these liquid units creates manifesting challenges

on Progress resupply missions. Backup systems, like the solid-fuel oxygen generators (SFOGs) in the service module, have been pressed into service. U.S.- and Russian-supplied stored oxygen provide a third leg of redundancy. In the near future, a new electrolysis-based U.S. oxygen generation system will be launched to the ISS to provide additional oxygen generation capability needed to support a six-person crew.

The carbon dioxide removal assembly (CDRA), in the U.S. segment, processes the cabin air to remove carbon dioxide, as does the "Vozdukh" system in the Russian segment. Failure of the desiccant containment, valve contamination, and corrosion resulted in some partial failures of the CDRA. However, by using new and innovative cleaning processes and by pre-positioning key spare components, the system was maintained throughout the shuttle down period. The advantage of having two totally different designs, one U.S. and one Russian, for carbon dioxide removal was evidenced.

In the wake of the *Columbia* accident, as logistics constrained the number of environmental samples being returned from ISS, the ISS Program reassessed minimum requirements for sampling. Environmental monitoring systems, such as the major constituents analyzer (MCA), have been used less frequently and for only the most critical measurements. When the volatile organics analyzer (VOA) failed, the United States and Russia shared returned air samples for analysis and monitoring of the cabin atmosphere. To reduce the number of environmental samples being returned, the crew performed previously unplanned microbiological measurements in situ to verify water quality. The important lesson for other long-duration missions is to critically assess and limit the number of environmental samples that must be returned, and to give the onboard crew the means to monitor and maintain the onboard environment.

When either a U.S. or Russian component has failed, the other country's system has always been relied upon for support. Despite the systems failures, multiple independent systems have proven complementary and have ensured maintenance of a safe, breathable atmosphere and a potable water supply. Dissimilar redundancy should be a strong consideration for Exploration systems.

Other systems also demonstrate the philosophical differences in design. The U.S.-provided health maintenance and exercise hardware is technically sophisticated, with vibration isolation and exercise performance monitoring systems, and provides excellent human and hardware performance data to the ground physicians and engineers. ISS has demonstrated their importance of resistive exercise systems for maintaining bone density and muscle tone.

The Russian-provided equipment is simpler and has limited monitoring or downlink capability, but it was based on systems that had been used for years on the Mir station and were specifically designed for simplicity, robustness, and on-orbit repair.

The sophisticated U.S. exercise hardware was not designed for on-orbit maintenance. The resistive exercise device (RED) and the treadmill were deemed non-critical early in the ISS program and therefore were not tested to the same extent as systems identified as critical. However, the U.S. exercise system hardware experienced failures soon after the first crew took up residence onboard. When requirements for the systems were investigated, it was determined that exercise is a critical need for long-duration missions and that the lack of adequate

crew exercise can threaten the mission. At the outset, the U.S. systems were designed for periodic return to Earth and replacement with new systems launched on the space shuttle. However, once on-orbit maintenance was deemed critical and with the challenges the program has faced in logistics, attention turned toward on-orbit maintenance by the crew.

In many instances it was found that the failed components of the exercise hardware are small. The crew has successfully been called upon to replace much smaller components than had ever been previously planned for repair in orbit. The maintenance operations necessitated some special zero-*g* considerations. For instance, the large gyroscope and flywheel of the treadmill vibration isolation system (TVIS) had to be disassembled from the treadmill assembly. The sophisticated vibration isolation and stabilization (VIS) system isolates the TVIS from the ISS structure, enabling crew members to run without transferring vibrations to the station or to sensitive experiments. On the ground this maintenance procedure is done on a workbench in a tightly controlled environment and with components resting on specially cleaned workbenches and with specially built restraints. In orbit, magnetic forces caused components to repel and fly away from one another. The crew members had to physically restrain the components and use considerable force to overcome magnetic forces during disassembly and reassembly.

The increased on-orbit maintenance requirements and the sometimes unanticipated maintenance difficulties have given great insight into the certification and testing requirements for hardware and into the kinds of operations astronauts can be relied upon to perform during long-duration exploration missions.

3. *Spacecraft Operations and Ground Support*

One of the major challenges for long-duration missions is the design of the ground support infrastructure needed to monitor and control the spacecraft systems.

Because the ISS is an international program, it faces unusual complexities in the area of real-time flight operations. Operations functions for ISS have been decentralized, with each partner taking on significant roles relating primarily to the hardware/systems they have developed. Over time, as the ISS moved into its operational phase, interdependencies have increased, and they will do so to an even greater extent in the future.

Real-time operations and control of the ISS are geographically distributed across countries and international partners, as depicted in Fig. 2. Each partner will eventually have an operations control center participating in flight operations, in addition to a launch control center for its transportation elements. Currently there are three control centers operating 24 hours per day, seven days per week, supporting the ISS: the Johnson Space Center (JSC) Mission Control Center (MCC) in Houston, Texas; MCC-Moscow; and the Payload Operations Center (POC) located at Marshall Space Flight Center (MSFC) in Huntsville, Alabama. The Mobile Servicing System (MSS) Operations Complex in Saint-Hubert, Quebec, supports operations of the Canadian robotics systems. The Columbus Control Center in Oberpfaffenhofen, Germany, and the JEM Control Center in Tsukuba, Japan, will come on line when the European and Japanese elements are launched. The control centers are interconnected, and each has its own unique functions and responsibilities. MCC-Houston and MCC-Moscow are responsible for the U.S.

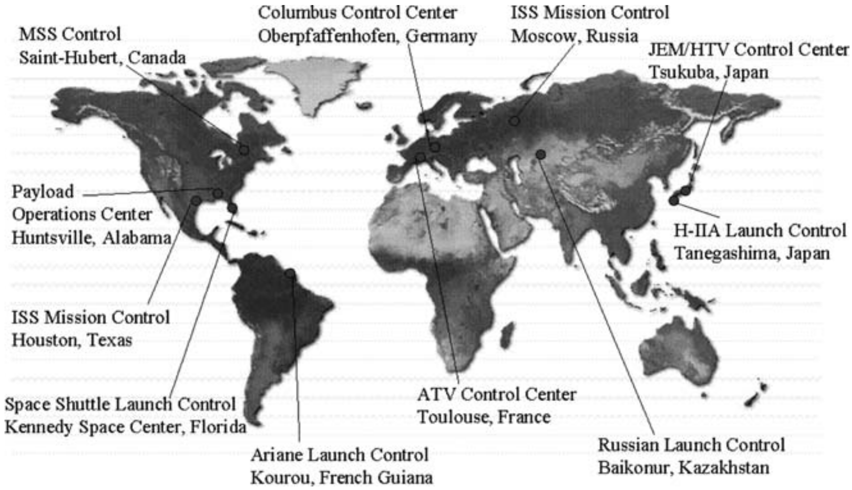


Fig. 2 ISS operations centers.

and Russian segments of the ISS, respectively. The POC is responsible for NASA payload operations, and generally falls under the authority of MCC-Houston. In addition to these prime control centers, there are also a variety of smaller operations centers supporting the research community. The prime control centers are not fully redundant, but have enough common functionality that they can provide backup capabilities in special circumstances. For example, when the MCC-Houston at JSC was evacuated due to approaching hurricanes, ISS control responsibilities were transferred to MCC-Moscow for the duration of the evacuation.

Long-duration, around-the-clock operations require a different approach than is used for short-duration missions such as the space shuttle. A prime consideration is the need to minimize overall costs for the ground support facilities and flight control teams. The ISS program has attempted to reduce mission operations costs wherever possible, without sacrificing mission safety or operational effectiveness. Efficiencies have been realized in both the ground support facilities and the flight control teams.

Another prime consideration for flight control team staffing on a long-duration mission is the human factors aspect. Long periods with shift or weekend work can disrupt family life, causing personnel burnout and high turnover. A variety of strategies have been employed to minimize these impacts to the flight control teams, such as reducing support on the weekends and off-shifts. Some flight teams cycle personnel on and off console. During the periods when they are not performing shift work, these controllers cycle back into planning or other operations support activities. The ISS prime shift hours were even driven by the very real constraint of the Russian flight controllers to utilize the Moscow mass transportation system, which does not operate from late evening to early morning. The exploration program will face some of the same challenges.

Reductions in the number of flight control personnel have been achieved by adopting different operational paradigms. At MCC-Houston on most weekends

when no critical activities are planned onboard ISS, only a single flight director and a few flight controllers may be working. The reduction in the number of support personnel means that flight controllers must be trained and proficient in more than one system to reduce manpower requirements and workload-induced burnout. The flight control team can also be reduced through increased automation for routine monitoring of spacecraft systems. Even more streamlining may be done in the future.

Another strategy is to reduce or simplify the "pre-mission" operations preparations activities that must be performed. A good example is the mission planning approach that has been adopted for the ISS. In contrast to the space shuttle program, where detailed flight plans are developed long in advance of the flight, the ISS program produces long-term plans at a much less detailed level. These plans allocate flight activities to days, but do not assign specific times. The very detailed flight plans are not generated until a week or two before they are to be executed. The template for generating the long-term plans has also been simplified. Initial concepts were to have three iterations of the plan. Over time, the ISS program has reduced the number of iterations, thus reducing the overall time and manpower required. Reductions in both the level of detail and the planning template have helped to minimize the manpower requirements for this activity.

The NASA ground support facilities continue to pursue reductions in sustaining costs, while increasing capabilities for the flight control teams. Both the MCC-Houston and the MSFC POC have been migrating legacy hardware/software to readily available and cheaper desktop systems, and have created internet versions of many basic flight control tools (e.g., voice distribution systems, information systems for flight support). This not only reduces facility sustaining costs, but allows more and more operations to be performed away from the control centers. MSFC has created a suite of low-cost tools, including the Telescience Resource Kit (TReK) and the Internet Voice Distribution System (IVODS), which enable U.S. science users to remotely monitor and command their payloads from their home sites. Because of the progress made in these remote operations support tools, MCC-Houston personnel were able to continue monitoring of ISS operations even after evacuating the MCC-Houston during recent hurricanes.

Flight operations concepts have accommodated the additional interfaces, complexities, and coordination that are introduced with multiple flight operations centers both within the United States and internationally. New tools have been developed to facilitate distributed planning and operations information distribution. Cooperative software development and sharing of software tools across NASA centers and partners, where feasible, has been used to reduce overall ground development costs.

Through these experiences, the ISS program has learned many valuable lessons in the areas of long-term flight operations, ground facility/software development, and distributed operations that will have applicability to the very complex and long-term exploration missions.

C. Crew-System Interface Operations

ISS has advanced robotic operations in space and demonstrated and validated the resulting human-machine robotic interactions and interfaces. The mixed crew

and robotic operations have enabled the extensive in-space assembly and orbital maintenance and repair operations. These same capabilities may enable and enhance lunar missions in the near term but will be a prerequisite for assembling and maintaining the large space vehicles that will be required to take people to the planets.

The Canadarm 2 robotic arm is used to assemble the large and massive ISS elements on orbit. Ground control of robotic activities enables more efficient use of valuable crew time. Development of displays and control for enabling astronaut control of these operations will be important for future spacecraft systems' designs. Software tools, such as virtual reality, play a role in helping crews to practice EVA or robotic tasks before ever donning a spacesuit or powering up the robotic arm.

The ISS provides a real world laboratory for logistics and maintenance concepts for future spacecraft. ISS crews have had to demonstrate repair capabilities as an indirect result of the *Columbia* accident and the reduced flow of logistics for the ISS. Crews and their ground maintenance counterparts have devised unique solutions that have kept the ISS functioning despite logistic shortfalls.

1. Systems Maintenance and Repair

The ISS program has demonstrated new capabilities to sustain spacecraft operations over long periods of time, which will be critical for lunar/planetary habitats and Mars transit vehicles.

The lifespan of hardware is frequently limited by performance and materials constraints. Hardware may be designed to remain in space without maintenance or replacement, designed for periodic maintenance or replacement, or designed with a specific certification lifespan. The ISS program uses a combination of analysis, testing, and simulations to define life limits. System performance is being tracked to understand the degradation of the vehicle and systems over time.

As a result of the shuttle loss and the resulting interruption of logistics support, all of these design features have been tested. Major challenges were posed by the limitations on size and mass of cargoes that could be launched to orbit, and by the inability to return failed hardware to the ground for failure analysis and refurbishment.

As discussed in Sec. III. A. 3, control moment gyroscopes (CMG) are used to maneuver the ISS. The system consists of four CMGs, although only three are currently required for full operation and two CMGs can provide adequate control. When CMG 1 malfunctioned, the remainder of the CMG system could continue to maintain vehicle attitude control. The cause of the failure of the rotational bearing of CMG 1 could not be resolved through telemetry transmitted to the ground. During the Discovery Return to Flight mission, astronauts conducted an EVA to replace the failed CMG, getting the system back into fully operating condition. The failed CMG was returned to the ground for failure analysis. Significant crew time and stowage volume was required to maintain hardware that was not designed for on-orbit repair.

An example of an external spare stowed inside a module due to environmental limitations is the bearing motor and roll ring module for a solar array. It is over 18 ft^3 in volume. After a solar array experienced several stalls in its rotation mechanism, shortly after the array was installed, the spare was launched. It was

determined that it would be too difficult to install without the space shuttle docked, and the shuttle has been grounded for three years. To date, the spare has been in storage for five years. For an exploration mission there will be limited stowage available for spares and no opportunity to return hardware for failure analysis, and so appropriate performance and diagnostic data must be available to support in-situ diagnosis and repair.

Maintenance tools must be available to the crew. Maintenance trades must optimize between complexity, automation, reliability, repair, and replacement. Factors that must be considered in the trades include crew training, crew time, stowage, logistics, costs, and vehicle functionality. Modular systems with commonality maximized across hardware and systems may be the best choice. The ISS is an ideal test bed for new maintenance methodologies and tools.

2. *Logistics, Resupply, and Stowage*

Resupply, logistics, and onboard stowage have proven to be very important issues for the space station. Prior to the *Columbia* accident, the nominal plan was to fly U.S.-provided consumable items as required on the space shuttle, and Russian-provided consumable items on the Progress cargo vehicle. This arrangement of frequent visiting vehicles provided a constant supply line that supported the crew and vehicle in orbit with less impact to on-orbit stowage. The Russian cargo vehicles were already planned to carry a significant volume of replacement components as the Russian hardware was designed for frequent maintenance. Critical U.S. hardware had pre-positioned spares, but all other hardware was to be flown on an as-needed basis. It was undesirable to pre-position all hardware on the space station due to the limited stowage space available.

Volumetric requirements for hardware and provisions stowage were addressed early in the ISS program, but ISS configuration changes introduced unforeseen challenges in the provision of adequate stowage volume. Careful attention by crew and ground planners has been required to ensure that access to emergency provisions, fire ports, and the module pressure shell can be maintained in case of contingencies, even as stowage has occupied many interior surfaces. Stowage usually occupies the volume of modules that are used less frequently, such as the Joint (U.S.) and Russian Airlocks and the pressurized mating adapter to which the shuttle docks. This incurs a penalty in terms of normal accessibility, difficulty in locating hardware and provisions, and increased crew time required to locate, unpack, and repack stowage areas.

In the closed and stowage-challenged spacecraft, inventory management gains critical significance. The items available onboard, their stowed locations, and their rate of use or lifespan must be tracked, forecast, and carefully planned. The computerized barcode inventory system used on the space station has been cumbersome in many instances. It is inefficient and maintaining the inventory demands considerable crew time. The recent apparent loss of critical Orlan EVA components onboard the station meant that the Russian EVA capability could not be relied upon until the components were located. Radio-frequency identifier devices (RFIDs) were explored early in the ISS program but decided against because of costs and technology availability, but the relatively low expense of

these now commercially available systems may be a prime candidate for use on lunar and planetary spacecraft.

In the wake of the *Columbia* accident, the resupply of the ISS depended on the limited capacity of Progress cargo vehicles. A food shortage could have necessitated abandonment of the ISS had a Progress not replenished the food supply. Out of necessity, the ISS program carefully reevaluated the usage rates and requirements for critical consumables of air, water, food, and propellant. The reduction in the resupply requirements allowed continued occupancy and operation of the ISS.

Good examples of resupply requirements reductions include a nearly 85% reduction in crew clothing, down from 12 ft^3 to just over 2 ft^3 per crew member for their six-month stay on orbit; a 25% reduction in food overage volume; replacing packing materials with soft goods such as towels and clothes; replacing film with digital cameras; using electronic procedures instead of paper procedures, etc. More water is recycled by fully drying out clothes and towels prior to disposal, which has led to a reduced usage from 3 to 2 liters per day per person for consumption and hygiene needs.

Propellant requirements were also reduced with careful planning. At 600 m^2, the U.S. solar arrays on the ISS are the largest power arrays ever flown. Each is nearly the size of a football field's end zone. These have the potential to create significant drag when oriented normal to the direction of flight. New array management techniques were devised to keep the minimum frontal area exposed while meeting required power demands. Such techniques include orienting the arrays' edges into the direction of flight when in darkness and holding the maximum possible bias toward that orientation during the sunlit portion of the flight. This strategy requires that some reduction in the electrical power be accepted. Different orientation techniques, termed Night Glider and Sun Slicer, were developed for the Earth-oriented and solar inertial flight attitudes, respectively. A third flight orientation, termed YVV, essentially flies the ISS sideways and results in extremely low drag. Overall, these techniques have resulted in over 25% reduction in atmospheric drag, and this has translated into a reduction of over 600 kg of fuel for reboost over a 30-month period.

With the conservation efforts of the crew and the close tracking of actual consumables usage, the program was able to maintain two crew members on orbit using only Progress cargo vehicles.

For exploration, the potential resupply of some items forces operational, philosophical, and hardware design trades. For example, should some food be grown to supplement the diet? How much trash and waste can be recycled? To what extent can you depend on a closed-loop regenerative water or oxygen life support system? Can clothes and soft goods packing or food packaging be made more efficient?

3. *In-Space Assembly Operations*

The size and complexity of the ISS presented a unique challenge to operations. The ISS at assembly complete will have a mass four times larger than any previous vehicle in orbit, and will be larger than a football field. The complexity, size, and mass of the on-orbit vehicle prevented assembly of the ISS on the ground. Even using a heavy-lift booster such as a Saturn V, many launch and

assembly flights would be required, but with launch capacity restricted to shuttle performance, a series of over 40 assembly flights was needed.

The elements that comprise the ISS are each between 10 and 20 metric tons, and with the exception of the Russian modules, each element is passive and must be carried to a rendezvous by the shuttle and berthed using a combination of the shuttle and ISS remote manipulator arms. While rendezvous between the ISS and shuttle was typically planned to be in coplanar orbits, when the planes were out of phase, this introduced new problems for such massive space vehicles that had to be carefully factored into rendezvous maneuvers. Similarly, the berthing of such large ISS elements needed to be very precisely aligned for the complementary berthing rings to mate properly. In the early years of the program, this was implemented with the use of a remote vision system, but later in the program, technology had improved to permit more precise alignments of elements using data from the shuttle and ISS inertial measurement units in combination with the pointing data from the ISS and shuttle manipulators.

The ISS international partnership introduced new challenges because elements and modules were designed and built by various international partners using the unique techniques and components of their respective countries and cultures. Also, the ISS is being assembled over an extended period of time. Some components will not be in orbit until 10 years after the launch of the first elements. Many of the components will have never interfaced with one another on the ground.

The first time the elements, including those produced by international partners, will be joined together and operated will be in orbit. The on-orbit construction of the ISS, starting from an initial single module, to the assembly complete configuration, will take over a decade; however, the ISS was required to be operational during all phases of construction. Not only did the major electrical power, thermal control, data management, environmental control, guidance, and propulsion systems need to be fully functional from the start, but the crew and research hardware was also functioning as the assembly program continued. And the ISS configuration is continually changing as additional elements are added and vehicles arrive, become part of the configuration, and then depart. The ISS is the first major human space system that was designed to be assembled, integrated, and operated in space by people, and only in space.

For the design of such a complex system, of paramount importance was to have complete understanding of the operations requirements throughout the life of the program. Often, this phase is “short changed” because of schedule and resource pressure or lack of experience on the part of the designer. For the ISS program, experienced engineers were allowed sufficient time and effort to analyze and understand the performance that would be required during the life of the program. Design options were investigated and the merits of each debated at length and preliminary analysis performed, before a design was selected. The importance of this effort cannot be overly emphasized.

The ISS design concept changed several times during the definition phase of the program. Together with the extended period of assembly and the complexities and inevitable problems that could occur over the assembly period, it was recognized that the ISS would need to be able to accommodate unforeseen changes.

The design required that control and operation of various systems and subsystems were distributed throughout the ISS. The system has several tiers

of modularity, at the component level, at the rack level, and at the module or element level. The U.S. segment of the ISS benefited from the establishment and adherence to this fundamental architectural principle. This most fundamental principle addressed hardware change-out and maintainability but required a system that was assemble-able.

As configurations and launch sequence planning have changed over the years, the modularity of the ISS architecture in the U.S. elements has proven critical. Modular racks with standard interfaces to the modules have allowed flexibility in manifesting and on-orbit outfitting. Racks were offloaded from the U.S. laboratory when the program changed the ISS to a higher inclination to accommodate the Russian launches. The modular architecture designed nearly two decades earlier allowed these changes in configurations and launch parameters with no impact to the hardware design.

The ISS is the first vehicle ever designed with rigid requirements for maintainability and re-configurability over a truly extended on-orbit lifetime. Each Apollo mission flew for only a matter of days and was used only once. Shuttles fly for a couple of weeks before they undergo major ground servicing and periodic major modifications. Skylab missions lasted for less than a year with no plan for continued use. Even Mir was designed for a five-year life, although it lasted for somewhat longer (15 years) through the addition of new modules.

The ISS was the first spacecraft ever to be physically assembled using extensive EVA and robotics in orbit. The shuttle-based assembly operations and missions are complex; almost every assembly mission is different. Earth orbit has become a construction site where conditions alternate between freezing cold and searing heat. The construction workers are extravehicular astronauts; the cranes are a new generation of space robotics; and the tools must operate in a microgravity environment.

Because of the complexity of ISS assembly, detailed assembly planning is crucial. Like an Earth-based construction site, certain activities must precede others, and so the integrated assembly sequence must consider all such dependencies. The ISS program plans and tracks the exact configuration of the ISS after each assembly stage, and ensures compatibility of the new elements into the existing on-orbit configuration. As new elements are brought on line, ISS documentation, software, procedures, operating plans, interfaces, and support tools are updated.

The only analog for future long-duration human exploration missions with modular spacecraft assembled in space is the ISS. Onboard systems and hardware are highly representative in design, complexity, and reliability to what will be required for trips to the planets. Many of the operational constraints are similar to those that will be experienced in the assembly of a lunar base or for a Mars spacecraft. ISS is the only test bed available today to check out systems and operations for Exploration.

4. Robotics

Efficiency, speed, and precision of the ISS in-space assembly requires that much of the assembly work be done robotically. ISS robotic systems are operated at the large-scale (e.g., cranes), mid-scale (e.g., anthropomorphic robots), and small-scale (dexterous and/or micromanipulators). Other reliable, remotely operated,

self-deploying and self-assembling systems were developed for use in Earth orbit, but are adaptable for use on the moon, and beyond. Intelligent and robust docking mechanisms, as well as autonomous rendezvous and docking technologies and the test beds used to develop them, are key exploration mechanisms.

ISS operations make use of an integrated suite of imaging sensors and manipulators for in-space assembly, inspection, and operation. The capability to perform a wide variety of local inspection and control operations will be important to the long-term, robust operation of diverse systems in deep space and on other worlds.

Canada, which built the space shuttle remote manipulator in the 1970s, also developed the station's primary mechanical arm. Called the Space Station Remote Manipulator System (SSRMS), the 55-foot-long arm has the capability to move around the station's exterior either like an inchworm, locking its free end on one of many special fixtures, called power and data grapple fixtures (PDGF), placed strategically around the station, and then detaching its other end and pivoting forward, or riding on a mobile servicing system (MSS) platform that will move on tracks along the length of the station's 350-ft truss, putting much of the station within grasp of the arm. Canada also is providing a new robotic hand for the SSRMS, the Special Purpose Dexterous Manipulator, also called Dextre. It consists of two small robotic arms that can be attached to the end of the main station arm to conduct more intricate maintenance tasks.

Two other robotic arms will eventually be installed on the ISS. A European robotic arm (ERA), built by the European Space Agency, will be used for maintenance on the Russian segment of the station, and the Japanese laboratory module will include a Japanese robotic arm that will tend research equipment mounted externally on a "back porch" of the lab.

These robotic systems introduce new techniques of human/machine interfaces. For example, training for the robotic operations is routinely performed with virtual trainers, and actual on-orbit operation is performed remotely by the crew using computer and television screens and assisted by an automated vision system.

A new mode of robotics operation was recently introduced on the ISS, when the MCC-Houston successfully performed a camera survey of portions of the ISS exterior, via ground control of the SSRMS. The onboard crew monitored the robotic operations, but did not actively assist in the event. The next day, the SSRMS was nominally moved via ground control to a clearance position to enable an upcoming Soyuz relocation.

Future programs will certainly utilize robotics extensively for assembly and maintenance tasks. The ISS experience with multiple robotic devices, human-machine interfaces, and crew robotics training should prove valuable.

IV. ISS as an Operations Test Bed for Exploration

The ISS affords a unique opportunity to serve as an operations test bed for the exploration tasks. Because it is a large, complex spacecraft operating continuously in space and maintained by the onboard crew, the ISS is an ideal platform to test protocols and procedures that will enable greater crew autonomy and reduce dependence on the ground support team. Training tools, crew and robotic

operations, time-delayed or intermittent ground communications, and on-orbit repair and maintenance can be demonstrated and validated in space. ISS can support demonstrations of new capabilities and tools required for sustaining spacecraft operations, including remote vehicle management, logistics management, in-space assembly and inspections, and flight demonstrations of new crew and cargo transportation vehicles.

Similar to spacecraft that will support future missions beyond low Earth orbit, ISS does not return to the ground for servicing, and provisioning of spares is severely constrained by transportation limits, especially after shuttle retirement. The ISS mission increments can be used as temporal and operational analogs for Mars transit. The ISS is a viable, and the only, test bed available in the near term for increasing technology readiness levels and/or validating concepts and technologies for human space flight in the microgravity, thermal, radiation, and contamination environments of space. It is the only space-based operational laboratory available for testing critical exploration spacecraft systems such as closed-loop life support, EVA suit components and assemblies, advanced batteries and energy storage, and automated rendezvous and docking. Table 1 describes some potential operations-related roles for the ISS as a test bed for operational experience and technology validation.

NASA is using the ISS as a laboratory for research with direct applications to Exploration requirements in human health and countermeasures, as well as applied physical science for fire prevention, detection and suppression, multiphase flow for propellant, life support, and thermal control applications. At the completion of assembly, the ISS will support research and technology development programs that meet the Agency's needs for crew health and safety, technology advancement, and validated operational experience essential for long-duration missions beyond low Earth orbit. With the transition to the Vision for Space Exploration, NASA's plans for research and utilization of the ISS have undergone significant changes. The resulting research and utilization approach is still evolving to focus available resources on risk reduction associated with the NASA exploration architecture. However, NASA is well positioned to take maximum advantage of the window of opportunity provided by the ISS.

V. Conclusion

The operation of the International Space Station was dependent at its outset very directly on the knowledge that was gained during operation of the earlier Russian and U.S. space systems. The ISS, as we operate it in space today, is an evolution of space systems technologies that were developed by many countries with widely differing design philosophies. The significance of the *Columbia* accident and its impact on continuing operations during the shuttle hiatus has been critical.

Many of the operations, processes, functions, and systems in use on the ISS today provide the capabilities that will be needed for future Exploration missions. Many of the hardware and software systems developed for ISS may be adapted for direct use on future systems. We are learning what is required to build and sustain a large space infrastructure over multiple generations. We have the test bed in place today to learn what does and does not work.

Table 1 ISS as an operational test bed for exploration

Mission objective	Capabilities needed	ISS role
Crew Operations: Crew operations and training	*For the moon* Integrated international crews Evolved operations tools and processes Skills-based intra-vehicular (IVA) and EVA training; evolved onboard training tools *For Mars* Integrated international crews Streamlined operations tools and processes Computer-based IVA and EVA training	Develop and demonstrate protocols and procedures with international crews Develop and demonstrate skills-based and onboard training tools
Crew Operations: Extra Vehicular Activity (EVA)	*For the Moon* Improved EVA suit materials and on-orbit maintainability Enhanced suit mobility/ flexibility; self-donning/doffing *For Mars* Highly reliable, maintainable suits; resilient to Mars dust Reduced crew preparation times for EVAs	Prototype new EVA suit materials, components and subassemblies Verify procedures for on-orbit repair and maintenance, self-donning/doffing, and airlock management
Spacecraft Systems Operations: Advanced Habitation and Life Support Operations	*For the Moon* Closed-loop life support Evolved medical care and countermeasures *For Mars* Long-duration crew accommodations Long-distance crew provisioning and resupply Advanced environmental control and life support Long-distance medical care and long-duration countermeasures	Evolve crew accommodations and planning systems for provisioning, food and clothing Characterize operating conditions for next generation closed-loop life support Validate advanced health care and countermeasures

(*Continued*)

Table 1 ISS as an operational test bed for exploration (*Continued*)

Mission objective	Capabilities needed	ISS role
Spacecraft Systems Operations: Communications Operations Protocols	*For the Moon* Remote systems management Systems monitoring tools for reduced ground support *For Mars* Remote systems management Radiation-hardened hardware Autonomous crew operations Autonomous systems monitoring tools	Develop operations procedures for remote vehicle management and intermittent communications Characterize operating conditions for radiation-hardened hardware and networks Validate autonomous crew operations and reduce ground support
Crew-System Interface Operations: Systems Maintenance; Repair; Logistics Resupply and Sparing	*For the Moon* Component commonality to support field repair without logistics resupply Reduced resupply requirements and trash generation Evolved logistics and inventory management *For Mars* Maximum component commonality to support on-orbit maintenance and repair Reduced in-route and on-site resupply requirements Autonomous logistics and inventory management tools	Demonstrate test, repair, and maintenance operations on orbit Evolve logistics management, maintenance and sparing concepts
Crew-System Interface Operations: Assembly Operations	*For the Moon* Reliable in-space assembly operations *For Mars* Autonomous in-space assembly operations	Demonstrate procedures for in-space assembly systems; self-deploying systems; inspection and control
Crew-System Interface Operations: Automation, Robotics and Human-Machine Interface	*For the Moon* Combined crew and robotic operations Robotic exploration aids and EVA support Ground-controlled robotic operations *For Mars* Autonomous crew and robotic operations with time-delayed communications Combined airlock and robotic operations	Validate robotic designs, concepts, tools and operational scenarios for long-distance assembly and maintenance tasks

Perhaps the most significant new operations knowledge gained from the ISS program to date includes:

1) New respect for the ability of the flight crew to independently perform system operations, maintenance, research, and mission planning, when provided with the appropriate training and tools to ensure greater crew autonomy.

2) The importance of the integral relationship between habitability, logistics, and crew physical and psychological support for long-duration missions.

3) The importance of designing equipment for longer lifetimes, with the capability to perform maintenance in space, and with an understanding of the certification criteria for continued use.

4) The complexity of the multinational and multi-organizational program management and operations, and the benefits as well as drawbacks these introduce.

5) Perhaps most significantly, the ISS program has educated a new generation of engineers and managers, providing first-hand experience in the design, development, integration, and operation of advanced human-operated spacecraft.

These valuable lessons should be factored into the exploration program from the beginning, so that they do not have to be relearned by the next generation of engineers and scientists who will take humans back to the moon and on to Mars.

Acknowledgments

This chapter would not have been possible without the contributions of the men and women of the International Space Station program and their supporting contractors. Additionally, the following individuals provided important contributions and support: Mark Uhran, Steven Pittotti, Andrew Thomas, Michael Foale, James Voss, Merri Sanchez, and John Bacon. Several individuals provided a thorough review: Gordon Ducote, Bruce Luna, Sean Fuller, Patrick Buzzard, and Ford Dillon. Any mistakes are those of the authors.

Bibliography

Carrasquillo, R. L., "ISS ECLSS Technology Evolution for Exploration," AIAA Paper 2005-0337, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 2005.

Chamberlain, J., Bradford, B., Best, S., and Nichols, K., "Utilization of Internet Protocol-Based Voice Systems in Remote Payload Operations," *Seventh International Symposium on Space Mission Operations and Ground Data Systems*, Houston, TX, Oct. 2002.

Fogleman, G., Leveton, L., and Charles, J., "The Bioastronautics Roadmap: A Risk Reduction Strategy for Human Exploration," AIAA Paper 2005-2526, Jan. 2005.

Fullerton, R. K., "Advanced EVA Roadmaps and Requirements," *31st International Conference on Environmental Systems*, Orlando, FL, July 2001.

Jones, W. R., "Observations of the Performance of the U.S. Laboratory Architecture," AIAA Paper 2002-4956, Oct. 2002.

Kitmacher, G. H., Rosenberg, I., and Gerstenmaier, W., "ISS: A Partner in Enabling Space Exploration Through Reduction of Risk," AIAA Paper 2005-2550, Jan. 2005.

Kitmacher, G. H., and Gerstenmaier, W., "The International Space Station, A Gateway to the Future," *International Astronautical Federation Conference*, Vancouver, British Columbia, Canada, Nov. 2004.

Kitmacher, G. H., "Design of the International Space Station Habitable Modules," *International Astronautical Federation, World Space Congress*, Houston, TX, Oct. 2002.

Maxwell, T. G., "Lessons Learned in Developing Multiple Distributed Planning Systems for the International Space Station," *Seventh International Symposium on Space Mission Operations and Ground Data Systems*, Houston, TX, Oct. 2002.

"The Vision for Space Exploration," NASA NP-2004-01-334-HQ, Feb. 2004.

Rhatigan, J. L., Robinson, J. A., and Sawin, C. F., "Exploration-Related Research on ISS: Connecting Science Results to Future Missions," NASA TP-2005-213166, Aug. 2005; also AIAA Paper 2006-344, 2006.

Sanchez, M. J., and Voss, J. S., "From ISS to the Moon, Mars, and Beyond—Applying Lessons Learned," AIAA Paper 2005-705, Jan. 2005.

Sellers, D., Pitts, L., and Bryant, B., "A Ground Systems Architecture Transition for a Distributed Operations System," *Seventh International Symposium on Space Mission Operations and Ground Data Systems*, Houston, TX, Oct. 2002.

Tsygankov, O. S., Alexandrov, A. P., Poleschuk, A. F., Abramov, I. P., Pushkar, O. D., and Grekov, N. S., "Russian-American Cooperation in EVA Area (from Russian Perspective)," *53rd International Astronautical Congress*, Houston, TX, Oct. 2002.

Walz, C., Uri, J., and Baumann, D., "Far from Home—Human Presence on the ISS as a Preparation for a Lunar Base and Beyond," AIAA Paper 2005-2549, Jan. 2005.

Chapter 26

Spitzer Space Telescope Sequencing Operations Software, Strategies, and Lessons Learned

David A. Bliss,* Vicken Voskanian,† and Timothy M. Weise‡

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

I. Introduction

THE Spitzer Space Telescope is the fourth and final of NASA's great observatories. Spitzer takes images and spectra in the infrared wavelengths of 3–180 μm. Spitzer (see Fig. 1) consists of a spacecraft, a 0.85-m telescope, and three very sensitive, cryogenically cooled science instruments that conduct observations in the infrared (Fig. 1). Spitzer's three cryogenically cooled instruments are the Infrared Array Camera (IRAC), the Infrared Spectrograph (IRS), and the Multiband Imaging Photometer for Spitzer (MIPS) (Fig. 1).

The IRAC conducts observations at near- and mid-infrared wavelengths. The IRS provides both high- and low-resolution spectroscopy at mid-infrared wavelengths. The MIPS provides imaging and some limited spectroscopic data at far-infrared wavelengths. Because infrared light is generated primarily by heat, keeping these instruments extremely cold makes them most sensitive to infrared light.

Launched from Cape Canaveral, Florida, on 25 August 2003, the mission was to be executed in three phases: a 60-day in-orbit checkout (IOC), followed by a 30-day science verification (SV), and a 2.5 year (minimum requirement) nominal operation. Spitzer's uplink process had to take on the tools, processes, and procedures required to support these mission phases while simultaneously meeting the mission's science viewing efficiency requirements.

---

*Spitzer Mission Sequence Team Lead.

†DAWN Mission Sequence Team Lead.

‡DAWN Spacecraft Team Lead.

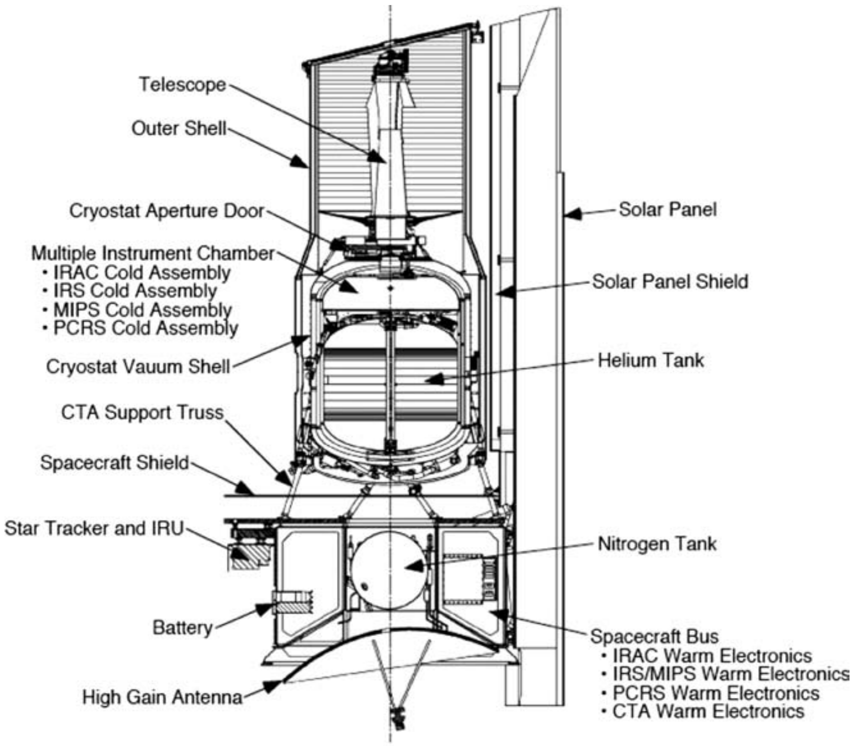


Fig. 1 Spitzer observatory.

A. Challenges

The three mission phases were distinctly different, and so strategies and sequence architecture would have to be developed to handle all three. Software, tools, processes, and procedures had to be able to handle all three mission phases, be adaptable enough to handle real-time updates during nominal operations, and most of all be able to achieve the mission's 90% science viewing efficiency requirement. [Science viewing efficiency is defined as time spent executing activities directly related to science observations. Spitzer's target goal is 90% science viewing efficiency. For each 24-h day, our goal is to spend approximately 21.5 h obtaining data directly related to science observations (engineering calibrations, the slew to Earth and downlink of data, idle time, etc., occurs during the other 2.5 h)]. To meet these challenges, Spitzer needed to develop a sequence architecture to support the IOC and the nominal operation phases and at the same time meet science viewing and spacecraft requirements. Spitzer needed to operate in a more non-deterministic manner to remove the inefficiencies built in by deterministic operations and slewing. Spitzer needed to generate the mission software, tools, processes, and the procedures to efficiently build mission sequences within this multiphase, highly efficient operations environment, and Spitzer needed to accomplish these objectives within project costs.

The first challenge had to do with the sequence types: if the sequences were going to be relative or absolute timed sequences; mini-, background, or stored sequences; and how the sequences were going to execute onboard the spacecraft. The second challenge was what kind of tools and processes we were going to require, and to be able to build these sequences with less hassle and in less time.

These challenges will be discussed later in this chapter, in more detail.

B. Background

The key to handling Spitzer's different mission phases and achieving its science viewing efficiency requirement was to optimize spacecraft flexibility, adaptability, and use of observation time. Thus the flight and uplink system was designed to accommodate non-deterministic spacecraft and science execution times. Much onboard flexibility was needed to load and run programs to achieve logical and event-oriented decisions that cannot be achieved using just absolute time-tagged calls. This onboard flexibility was achieved by using virtual machines to enable a multi-engine onboard sequence architecture. Each virtual machine represented one sequence engine and executes time-tagged instructions capable of invoking complex functions.

A virtual machine (VM) provides the basics of a computing environment. Each virtual machine employs simulated memory locations, simulated registers, a stack pointer, and an instruction pointer. Each VM has code space that contains storage for the sequence loads, blocks, or other software modules, and execution space in which the execution of the instructions takes place. The VM strategy includes using a language [Virtual Machine Language (VML)] to add logic to sequences to utilize this architecture. It should also be noted that the VM architecture has limitations such as sequence load size, instruction limit, and number of parameters being passed to a block. Adopting this multi-engine VM architecture for the underlying sequence capability simplifies software design and provides a great deal of flexibility for creating and running programs [1].

The number of virtual machines is sized to address the need for simultaneous threads of execution. One thread of execution may run per virtual machine, but an arbitrary number of functions may be run on any one machine. Cross-machine function calls maintain pointer and stack information on the calling virtual machine. Global data are visible to all functions and sequences.

Spitzer eventually selected 12 virtual machines to address the anticipated needs for simultaneous threads of execution. One thread of execution may run per virtual machine. The time clock in each Spitzer VM updates every 0.1 s.

II. Lessons Learned

A. Sequence Architecture

Spitzer's multi-engine virtual machine architecture is unique. The "master" sequence is built as an absolute-timed file that controls the behavior of the overall sequence load by spawning relative-timed or calling absolute-timed slave sequences at appropriate times. The master sequence also calls absolute-timed engineering calibrations and downlink activities. There may be one or more master

sequences per week, but only one master may be executing at a time. Slave sequences may contain all the commands to execute their activities, or they may call blocks from the block library, or they may call one or more astronomical observatory records (AORs), instrument engineering records (IERs), spacecraft engineering records (SERs), or other activities chained together. If the slave and its associated AOR(s), IER(s), and SER(s) will not fit into one VM module, the slave will call one or more slave library. The slave library contains only single-use activities at the AOR/IER level. As with the slaves, the slave library issues AOR(s) and IER(s) chained together. Spitzer also uses a set of functions called sequence blocks, which are parameterized, reusable relative sequences. Parameterization allows execution of the block to occur differently with each use. Blocks may accept parameters, return values, or both. Blocks are loaded onboard and may be used by the master or slave sequences in AORs, IERs, or SERs. As mentioned, VM engines are used for two distinct purposes: one is to store functions and data, and the other one is to execute threads of processing (Fig. 2). Functions can only be stored on one engine and may be executed on multiple engines simultaneously. Slave sequences may contain all the commanding necessary for the execution of their activities or they may call blocks from the block library and slaves from the slave library (Fig. 3) [1].

A sequence engine, or VM, can be loaded with a sequence, and can execute while other VMs are being loaded and executed. Figure 2 shows Spitzer's 12-VM sequencing architecture.

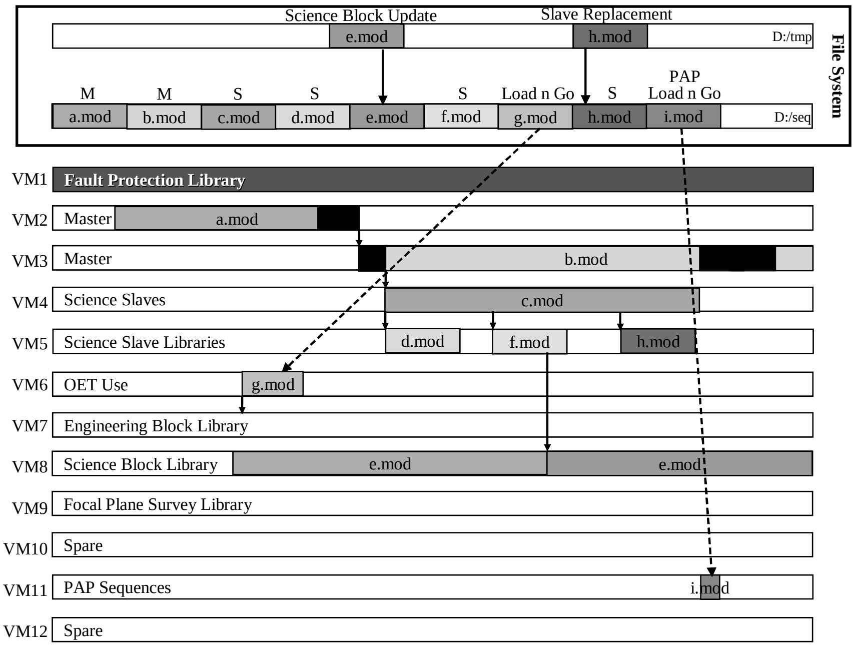


Fig. 2 Twelve-VM architecture.

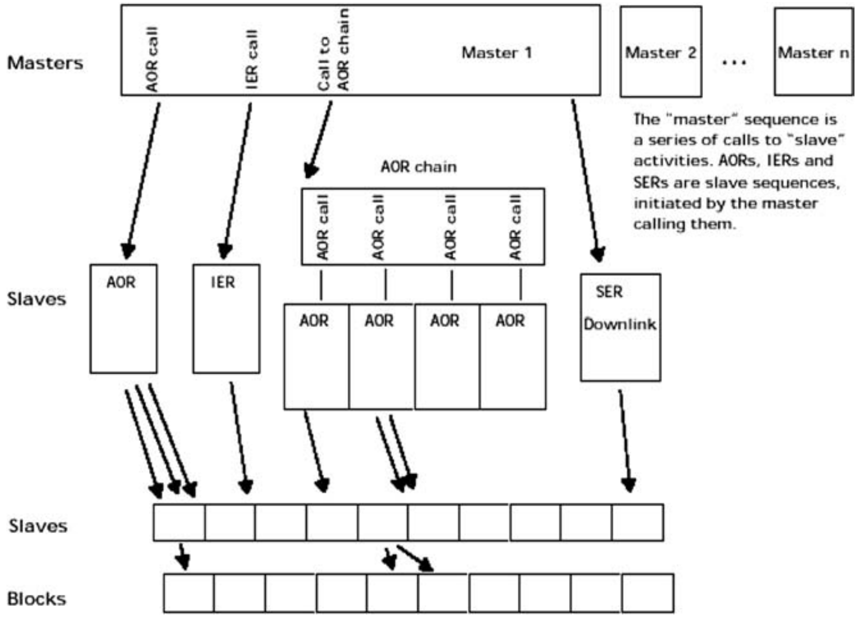


Fig. 3 Master–slave sequence architecture.

As an example, let us consider how the architecture would work to execute a nominal science load (a more detailed discussion of nominal science operations is included later). The master sequence is loaded on virtual machine 2 or 3 (for this example, let's select virtual machine 2). The absolute-timed master sequence plays operations cop for the week. It executes absolute-timed engineering events as required for spacecraft safety and downlinks as Deep Space Network (DSN) viewperiods become available. Between these absolute-timed events are periods of autonomous operation (PAOs). Each PAO is approximately 11.5–12 h long and is constructed mostly of relative-timed slave sequence(s) that issue commands, relative-timed slave libraries, AOR, IER, and SER activities, or stored onboard blocks. The master spawns the PAO slave sequence(s) off to virtual machine 4. The first activity in the slave begins, calling an AOR from the slave library that in turn calls a block. The Pointing Control System (PCS) slew complete indicator global variable tells these activities when observatory slews are completed and observing can begin. By studying statistical variations in the slews, the Spitzer operations teams have been able to update the slew model to better predict the exact time for slew completion.

The slave or slave library activities (AORs, IERs, and SERs) are tied together so that the second activity will start as soon as the first activity is completed. When the first activity completes, the next activity starts, calling its own blocks, etc.

A slave concludes with one of three prevailing conditions. In the first condition, the slave will complete before the next slave (or downlink if at the end of the PAO)

begins and there will be “dead” time. The only negative result for this condition is that viewing efficiency will not be completely optimized. In the second condition, the slave will complete as predicted, and the next slave (or downlink activity) will start immediately. This condition is optimal in that no science data are lost and viewing efficiency is optimized. In the third condition, the slave will not have completed and will need to be killed before the next slave (or downlink) can start. In this condition, statistical variations have worked against you, and the slews in the slave average longer that expected. When a slave is killed, a halt command kills the executing slave sequence and performs a clean up before the downlink activity starts.

For downlink, Spitzer turns its high-gain antenna (HGA) to Earth and dumps its science and engineering data and possibly its reaction wheel momentum data as well. When the contact is complete, the master sequence controls the return to observing by spawning another slave sequence onto VM 4.

Near the end of the week, the next master sequence and its associated slaves and slave libraries are uplinked. As its last activity before completion, the current master sequence kicks off the next master sequence onto VM 3 and then expires.

1. *Requirements*

In addition to building sequences that would handle Spitzer’s different mission phases while achieving its science viewing efficiency requirements, Spitzer had to handle several types of sequences. These sequence types include mini-, master, absolute, and relative-timed sequences. Spitzer also had to handle special events like Target of Opportunity and Slave Replacements. As mentioned, to handle the master and slave sequence structure, the VM was introduced to solve storage and sequence execution issues. Adaptation of existing, core mission software enabled reliable and cost-efficient generation of sequences. In addition, considerable time and effort was spent building scripts to support the sequence builds. This script development was not anticipated at the project’s beginning. During the IOC phase, the project required continuous coverage to enable ground-in-the-loop operations and to ensure that the amount of science being gathered could be downlinked before the mass memory card (MMC) filled up.

2. *Pros and Cons*

The sequence design and ground system developed to support the launch, IOC, and the primary mission presented challenges with the sequence design and with the ground system. The multi-engine sequence architecture was the major role player. The sequence architecture was designed to support absolute- and relative-timed sequence, where these sequences could execute in different engines. At that time, Spitzer was the project with most extensive use of the VML. The VML was used as a source language for spacecraft sequences. One advantage to that is the uplink volume reduction (order of magnitude). It also increased science efficiency through relative timing, non-deterministic waits that reduce using worst-case timing (may kill activity if it is not done on time before the next absolute-timed event). The uplink size was reduced since the blocks reside onboard. It also gave flexibility in sequence, meaning that sequence and block development can start

earlier and proceed in parallel with spacecraft development. Also it reduced the flight system maintenance cost since many changes occur in blocks rather than flight software. At the same time there were some disadvantages, too, when it came to the VML. There were some difficulties because of integrating VML with the existing Jet Propulsion Laboratory (JPL) legacy systems such as the Sequence Software Adaptation Team (SEQ). Translators were needed and tools had to be generated by the teams to be able to support the project, especially when it came to the sequence builds. The teams had to generate a number of tools to be able to support the sequence builds. Limitations in ground hardware and software have also played a role. By not having the right hardware and the right software, and the sequences being too big, the sequence generation process was taking a tremendous amount of time.

B. Sequence Development

Spitzer Science Center (SSC) uses the software package SIRPASS to develop sequences from the database of AORs, IERs, and SERs. SIRPASS uses the database and metadata to create the one-week schedule from a pool of observations. SIRPASS controls oversubscription and sequence packaging. Once SIRPASS has created the schedule, it outputs the master SASF, at least one SATF for each slave sequence and for each slave library. The SSC also takes engineering files from the OET, which are called in absolute time from the master sequence. These products are then delivered to the Spitzer Mission Sequence Team (MST) for command and modeling product development.

The MST then uses SEQGEN to merge the SASF and SATFs from SIRPASS with the engineering SATFs from LMMS OET. MST uses SEQGEN, SLINC, and the VML compiler to convert these sequence components into spacecraft readable language, and produces review products and products to instruct the DSN. The project then reviews the sequence products and submits comments. Each sequence generation and product review cycle is called a “pass.”

The nominal science sequence schedule allows for two passes. The second pass is usually intended to fix problems discovered during the first pass. Other sequence processes that are more tightly constrained in time, such a Target of Opportunity (TOO) or slave replacement, allow for only a single pass.

The process of generating command and modeled products is divided into three subprocesses: command product generation, modeled and review product generation, and Deep Space Network instruction product generation. Additional products to support mission sequence development are attributable as secondary processes.

To perform Spitzer sequencing, we applied Spitzer mission-specific adaptation to certain core mission programs (SEQGEN, SLINC, CMD_TCWRAP, etc.) maintained by JPL’s Mission Management Office. Spitzer mission-specific adaptations were enabled by adaptations made to mission-specific “adaptation” files that define the mission-specific commands, models, and constraint checks.

As an example, let us consider SEQGEN. SEQGEN allows a user to generate and modify requests, expand a series of requests into their resultant spacecraft (S/C) commands, model these S/C commands, flag conflicts in the modeling of commands, flag violations of flight/mission rules, show the time extent of each request graphically, and graphically display model attributes. As implied previously,

SEQGEN consists of a multimission core program and a mission-specific adaptation. The mission-specific adaptation employs the following set of adaptation files that define the mission-specific commands, models, and constraint checks:

1) Spacecraft Model File (SMF). The SMF contains the definition of spacecraft and ground subsystem models, and spacecraft command/parameter definitions.

2) Flight/Mission Rules File (FMRF). The FMRF contains flight and mission rule checking algorithms.

3) Spacecraft Activity Type File (SATF). Contains names and definitions of the activity types, including onboard blocks, ground expanded blocks, SEQGEN directives, and SLINC directives.

4) Context Variable (Definition) File (CVF). Contains parameters defined during the adaptation process that are used in the definition of activity types or models.

5) Legend File. Contains data to define display definitions and layout.

SEQGEN requires the following input files to perform sequence expansion and constraint checking: 1) Spacecraft Clock Coefficient File (SCLK); 2) Lighttime File (LTF); 3) DSN Viewperiod File (VP); 4) Viewperiod Format Description File (VIEW_FD); 5) DSN Station Allocation File (SAF); 6) Initial Conditions File (INCON); 7) Context Variable File (CVF); 8) Spacecraft Activity Sequence File (SASF); and 9) Spacecraft Activity Type File (SATF).

SEQGEN outputs are the following: 1) Spacecraft Activity Sequence File (SASF); 2) Spacecraft Sequence File (SSF); 3) Predicted Events File (PEF); 4) Final Conditions File (FINCON); and 5) Run Log.

Figure 4 [2] shows SEQGEN inputs and outputs. In both cases, note the input of the SMF, FMRM, and SATF files “adaptation” files. Figure 5 [2] shows the overall flow chart for the set of core Mission Services and Applications (MSA) software, which is adaptable to support multiple missions.

In addition, these adaptable, core tools have been “wrapped” so that their use enables a consistent, multi-mission process. These “wrappers” are scripts that use tables that define states for different spacecraft. Spitzer was incorporated into these wrappers. For example, when performing sequence integration, the Mission Planning and Sequencing Team (MPST) user identifies Spitzer’s spacecraft number. The script then references Spitzer’s spacecraft data. These data tables then tell the script what to do for Spitzer. Thus if a user is running SLINC, the script will use Spitzer’s spacecraft number to consult a table that identifies Spitzer as a VML spacecraft. The script then runs a VML compiler as opposed to a memory management routine that is executed for a non-VML spacecraft.

Generating and reviewing sequence products for a multi-engine sequence architecture entails much work. Early in the development of the MST process, it became clear that scripts would be needed to work in unison with adapted core software to make timely sequence development possible. As an example, MST developed a sequence command processing script to build command products for each of the master, slave, and slave library engines, and then merge these individual engine command products into one, integrated command sequence.

To build a one-week science upload sequence, each sequence (master, slave, slave library) must be individually processed into command packet files.

For each week of normal operations, SSC delivers to MST one absolute-timed master SASF file, multiple slave SATF files, support AOR and IER SATF files,

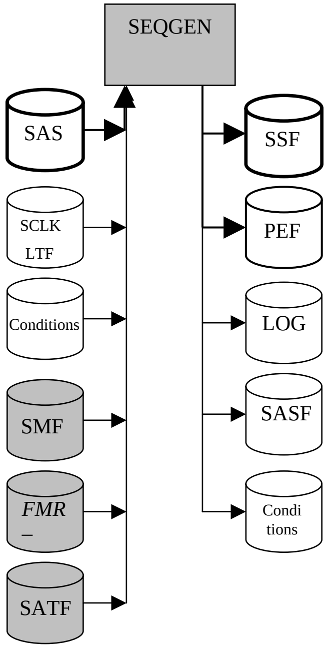


Fig. 4 Uplink data flow for surveyor bus.

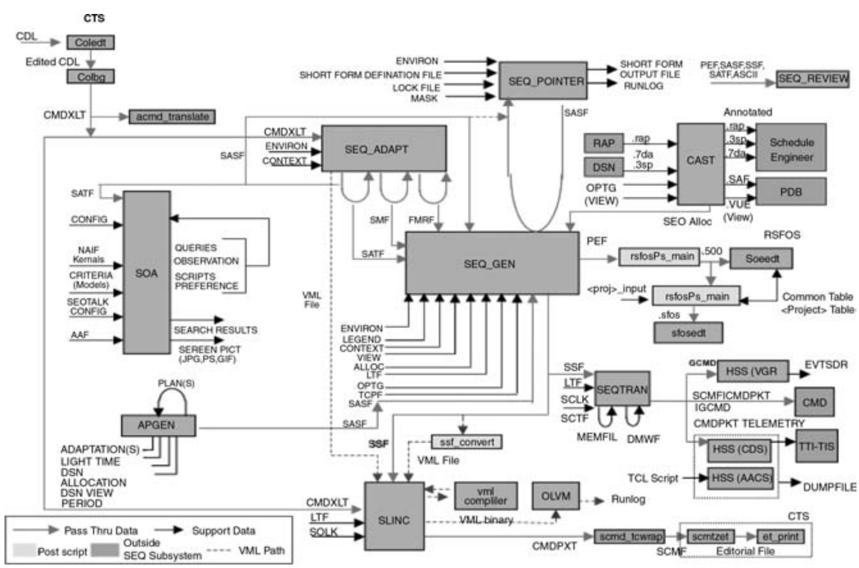


Fig. 5 MSA software.

and an SPDF file that details the overall sequence structure (i.e., how many slave libraries will each slave sequence call, and how many support SATFs will be contained within each slave library).

For example, for a nominal science sequence, each master sequence, each slave sequence, and each library slave sequence must be individually constructed, translated into spacecraft-readable code and merged with all the other master, slave, and slave library sequences into a single, week-long command sequence. To perform this process by hand would prove extremely labor intensive and make the weekly production of science sequences virtually impossible. Thus MST developed a sequence command processing script to automate this process. In performing its functions, the script executes the manual steps involved in assembling the various pieces of each sequence, executes more multimission software processing and processing checks on each sequence, and outputs each master, slave, and slave library's command products. The software then merges these command strings into a final, integrated command sequence. This single, automated command processing routine potentially saves days of labor for each nominal science sequence.

Multimission adaptation of core software, the development of scripts that automated manual processes and interacted with core software, and streamlined operational processes have helped Spitzer achieve its mission objectives. Spitzer mission operations have been cost effective and efficient. However, development and testing of the mission sequence flight software architecture were not easy. In addition, the non-deterministic slew modeling was challenging. The flight software algorithms had to be incorporated into the ground software so that spacecraft motion could be exactly modeled on the ground. If a project is willing to accept truncation of relative-timed slave sequences, non-deterministic modeling could be dropped, thereby simplifying ground operations.

1. *Test and Training Sequences*

Prior to launch, test sequences were generated to validate blocks, flight rules, and sequence formats. The test sequences also helped to train the teams on how to build sequences and to get ready for operations. The test sequences were then delivered to the test bed for simulation. Initially, we developed sequences to test the commanding of single instruments. Then we developed sequences to test commanding to all three instruments within the same sequence. These multi-instrument sequences included instrument transitions since, for Spitzer, only one instrument can operate at a time. Once command products were successfully produced, the sequence would be loaded onto the test bed for checkout.

A number of Operation Readiness Tests (ORTs) were conducted to test personnel readiness for operations. Although sequence build inputs were often delivered late and contained problems, they established the groundwork for subsequent training builds. With time, the teams became more proficient in producing and delivering products. Test and training played a major role in getting the teams ready for operation. Several types of sequences were introduced during the development process for each phase of the mission. For IOC, mini-type sequence was introduced, where the sequences were relative timed and they were designed for

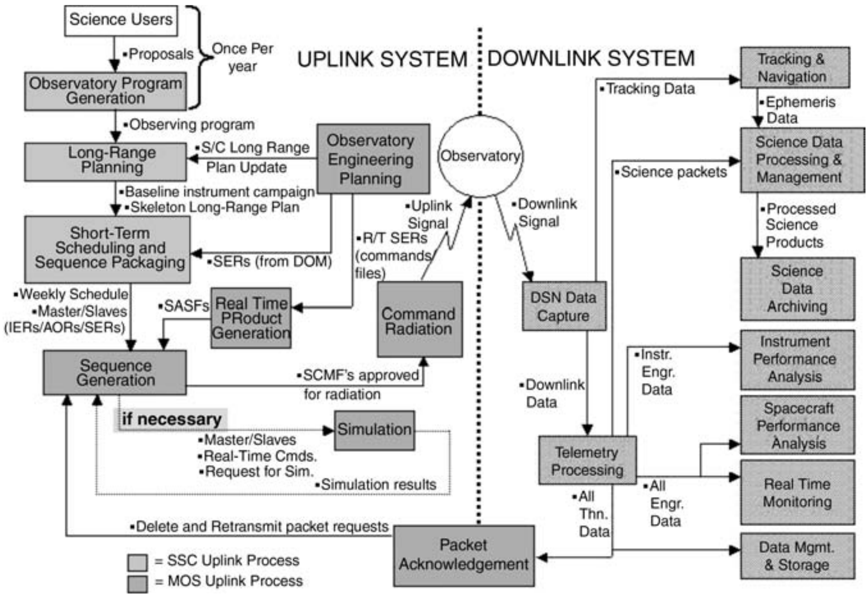


Fig. 6 Spitzer operations system.

specific campaigns (instruments). Test and training sequences generally used a single pass of process steps 1–8 outlined previously.

2. *In-Orbit Checkout/Science Verification*

The Spitzer IOC and SV phases were planned to last 90 days. IOC was planned to begin 5 h, 42 min after launch. SV was planned to begin 60 days after launch. The IOC and SV Mission Plan were introduced to commission Spitzer for routine operations. The emphasis for the IOC phase was to bring the facility on line safely; verify the functionality of the instruments, telescope, and spacecraft; and demonstrate that the facility meets the level 1 requirements. The emphasis of the SV phase was to characterize the observatory in-orbit performance, demonstrate observatory capability for autonomous operations, conduct early release observations, and exercise the ground systems software, processes, and staffing sufficiently to commission the facility for routine operations. During the IOC phase final focusing was achieved, and the telescope had cooled to an operating temperature of approximately 5 K (−268°C or −451°F). This cold temperature has allowed the observatory to detect the infrared radiation, or heat, from celestial objects without picking up its own infrared signature.

The IOC was completed in 62.8 days and SV in 35.6 days. The longer duration was due primarily to three safing/standby events experienced by the spacecraft. IOC sequences were built by instrument campaign (each campaign was a mini-sequence and could be from a few hours to a day or so in duration), which allowed more flexibility to respond to changes based on the new data. Because IOC/SV

sequences were event driven, a flexible, real-time strategy was selected. Background master sequence loads were 1–2 weeks in duration with the instrument campaigns loaded into the sequence engines using the “load and go” approach with relative-timed sequencing.

In IOC, the OET submitted the absolute-timed master SASF while the relative-timed mini sequences from the SSC. MST modeled the master with planned mini-sequence execution times to insure successful integration and flight rules compliance.

IOC/SV phases confirmed the following:

1) Relative-timed sequencing is the key to flexibility.

2) Reserve time should be distributed throughout a timeline to ensure that a complex, interleaved set of dependent activities is robust against unplanned anomalies.

3) Tools must have a simple way to model dependencies, since the planning process must allow a rapid, frequent response to changes and anomalies.

4) Allow for frequent communication between teams and team members. The replan team required all disciplines to be able to recommend solutions at a rapid pace. The IOC/SV web site with access to all important documentation, tools, forms, useful links, and daily status was one-stop shopping for the distributed teams.

5) Quick replanning allowed for other teams downstream the standard amount of time to do their jobs.

6) Test and training exercises provided invaluable experience. Spitzer performed eight replan training exercises prior to launch, starting simple and ending with more challenging situations. Tabletop training exercises were chosen to exercise the team’s decision-making capabilities.

7) Phased transition from IOC to SV to nominal operations allowed the teams to get up to speed smoothly. We had team responsibilities changing as well as the change from relative-timed sequencing to absolute-timed sequencing. The IOC replan team was kept on during transition.

8) Co-location, clear lines of authority and responsibility, and the fact that the key IOC team members had no other project responsibilities during that period were also key factors in the success of IOC/SV.

3. *Nominal Science Operations*

The transition to nominal operations occurred right after IOC/SV. The IOC/SV process was designed to allow for short lead times, while the normal process involves almost six weeks of development time. It took some time (approximately 12 weeks) before the project fully achieved the standard operations.

The prime mission started with a fully commissioned facility. The prime mission extended from the end of SV until the liquid helium supply was exhausted. An extended mission would take advantage of the short wavelength IRAC bands to continue useful observations. End of mission is expected at ~ launch + 5 years (note the requirement is 2.5 years).

Nominal operations are being used to operate the facility at the start of the prime mission. During nominal operations, the Mission Operations System (MOS) commands the spacecraft systems and instruments to gather science data, telemeter that data to the ground where it is initially processed by the MOS, and then passed on to the SSC. DSN tracking using the 34-m and 70-m antennas

nominally occur once or twice a day with each pass approximately 12 h and 24 h apart, respectively. Engineering and science data are nominally being transmitted by the facility on the high-gain antenna at 2.2 Mbps. As the Earth-sun-spacecraft geometry worsens, the high-gain antenna data rate will drop but the project can mitigate this by switching to a downlink configuration where both power amplifiers are turned on to increase the downlink signal strength or switch to 70-m station coverage.

The strategy for accomplishing the prime mission was well planned and rehearsed. Sequences of commands were designed, built, tested, uplinked to the observatory, and stored onboard for later execution. Operations personnel had been prepared for nominal operations by walkthroughs, rehearsals, Operational Readiness Tests, and by operating the observatory during IOC/SV. Mitigation plans, and contingency sequences for a select set of contingencies, were prepared prior to nominal operations and continue to be maintained.

Nominal operations are more unique. Nominal operations start off with science observation proposals, which the science community submits to SSC approximately once a year. These proposals are collected into an observing program. This observing program is used for long-range mission planning, for developing a skeleton long-range plan, and for developing a baseline instrument campaign. These products are used for short-term scheduling to build a weekly schedule. This weekly schedule is packaged into products (a master sequence and its corresponding set of slave sequences) that are used by the MOS to build uplinkable sequence loads.

The MOS builds sequence loads of approximately seven days in duration and radiates them to the observatory for execution at a specified time. Sequence loads become active at a set time, execute to completion, and then are replaced by another sequence load of similar length, thus there will nominally be a sequence load active at all times on the observatory.

As mentioned, sequence loads will take advantage of onboard blocks that can be executed at preset times. These blocks contain a series of relative time-tagged commands that perform a particular function. Parameters can pass data to onboard blocks, allowing these blocks to be uploaded once and used multiple times.

The MOS also has the ability to uplink any individual observatory command. The command list in the Command and Telemetry Dictionary indicates whether a command can only be sent as an immediate command (real-time command), only as part of a sequence load or “block,” or both.

As also mentioned previously, sequence loads execute onboard the observatory in VMs. Spitzer uses a 30-calendar-day nominal sequence load uplink process from weekly schedule generation to uplink of the sequence load with a 30% development margin. This cycle allows five days (10 nominal DSN passes, including 30% margin) to get a sequence loaded onboard the observatory. Note that there is work done at the SSC by the operations teams and the instrument teams prior to schedule generation that includes pooling of the available observation proposals for scheduling, and generating calibration strategies for the software interface specification (SIS).

The MOS radiates a merged Spacecraft Message File (SCMF) of the master and slave sequences. Large uplink volumes may require SCMFs to be split to fit within scheduled DSN passes.

As mentioned earlier, the sequencing strategy chosen for Spitzer employs a master/slave sequence architecture. The master sequence controls the behavior of the overall sequence load, spawning slave sequences at specified times with the slave sequences issuing commands, AORs, IERs, SERs, and sequence blocks.

In addition, as the flight software dynamically changes the values of global variables, executing sequences and blocks can make real-time decisions based on the values of these variables.

During nominal operations the master and slave sequences perform their own activities. Master sequences contain activities to, among other things, load slaves into sequence engine 4 and to call, spawn, or halt slave sequences.

Slave sequences contain science observations and instrument calibrations. As mentioned before, slave sequences perform engineering functions that, among other things, perform command history "dumps," load slave libraries into sequence engine 5, end downlink passes by calling the "stop downlink" block, and execute all science IERs and AORs as well as engineering SERs.

Sequence blocks are small pieces of sequence code (observatory commands and VML instructions) that reside in libraries. Blocks are used in developing sequence loads to save code space (i.e., uplink volume) in a VM or to execute repetitive and routine activities onboard the observatory. Blocks can be used to implement science or engineering activities. Spitzer has several block libraries. As shown in Fig. 3, the science fault protection library is resident in VM 1. The engineering block library is resident in VM 7, and the science block library is resident in VM 8.

Blocks can be passed parameter values from a sequence to customize the block execution, and blocks can return values. For the programming experienced reader, blocks are similar to subroutines or functions that can be called from a main program.

Sequences may call blocks as library routines or to arbitrate a resource. Blocks may also call or spawn blocks, but blocks may not call sequences.

When a block is spawned, the block executes on a different engine than the calling sequence, and thus the calling sequence does not wait for the block to complete.

When a block is called, the block executes in the same engine as the calling sequence, and thus the calling sequence must wait for the block to complete execution.

4. *Real-Time Operations*

Real-time operation includes monitoring health and status of the observatory, monitoring the status of DSN antennas, the telemetry, and command processing capabilities. The real-time command process is a subset of the uplink process, and does not follow upon the stored sequence process are Express. Non-interactive real-time commands and files do not impact onboard resources. These files and commands can be processed quickly and do not require flight rules checks. They do not require mission manager approval before uplink. Interactive real-time commands and files impact onboard resources. They can be processed quickly, but may require flight rule and constraint checks and always require mission manager approval before uplink.

Real-time commands are used by the OET to implement engineering activities during track times. Unless pre-approved, all real-time commands require the originator to submit a change request prior to building and uplinking the command.

Real-time commands provide an advantage in that some of them are pre-built and can be reused. However, too much real-time commanding can produce an undisciplined "joy sticking" of the observatory.

5. *Target of Opportunity*

Target of Opportunity (TOO) observations are observations approved during the general observer review process, but which involve transient phenomena whose exact timing and/or location on the sky were uncertain at the time the proposal was submitted [e.g., a newly discovered comet, a bright supernova, or a gamma-ray burst (GRB)]. TOOs are categorized by the extent to which the execution of such an observation affects normal scheduling and observing procedures. A *high-impact* TOO is one with a delay of less than one week (minimum of 48 h). A *medium-impact* TOO is one with user-specified delays of one to five weeks. A *low-impact* TOO is one where the acceptable delay is longer than five weeks. All delays are measured from the time the SSC director approves the TOO activation request until the time the first observation in the newly approved TOO sequence begins execution on the observatory.

The TOO can be achieved by doing a master replacement or a slave replacement. The master replacement is a high-impact TOO, often involves instrument transitions, and requires complete regeneration and remodeling of a replacement sequence. The executing sequence must be stopped and deleted, and the replacement sequence loaded and activated. The subsequent sequence must also be remodeled to ensure that it transitions smoothly from the replacement sequence. The TOO master replacement is a single pass process (see Fig. 7) and is allocated 48 h to complete. However, we have performed the TOO master replacement in 36 h.

The slave replacement TOO is of medium or low impact. If we are replacing a slave or set of slaves for an uplinked ore executing sequence, the impact is medium. The new slave(s) must be generated and the executing sequence remodeled with the new slave(s). Once reviewed and found acceptable, the new slave(s) are uploaded and the onboard slaves they are replacing are deleted. Since the new slave(s) must be onboard before the master calls it (them), turnaround time is short. The low-impact TOO applies to slave replacements for a sequence that has not been uplinked. It usually can be accomplished with a single, additional pass.

6. *Anomaly Recovery*

The Spitzer mission has had few anomalies. However, there have been a couple of safe mode and standby events that required rapid response and a high degree of coordination between the MOS teams to get the observatory back on line quickly. During such an event, the flight control team communicates with the spacecraft, by gathering as much data as possible, which, under these circumstances, will be broadcasting via the low-gain antenna, while the OET and others attempt to diagnose the problem and come up with a solution. Once the OET has a good estimate

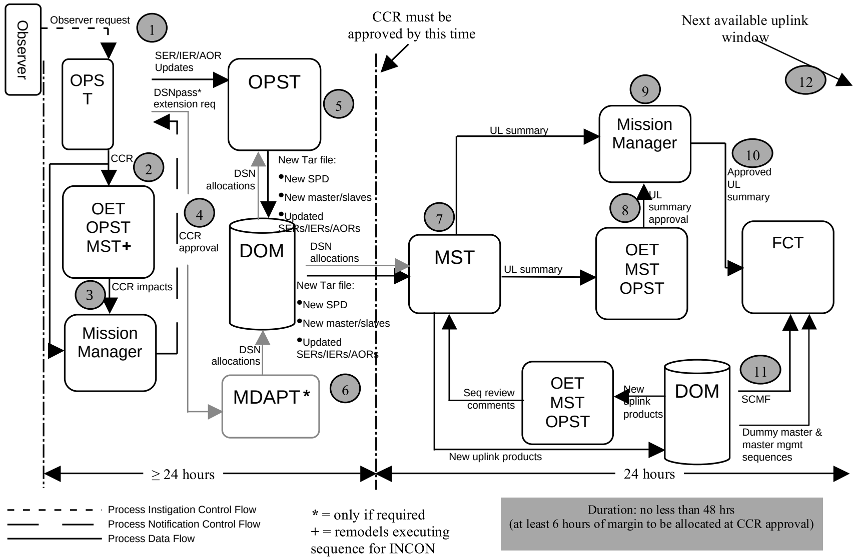


Fig. 7 Master Replacement TOO.

of when they will begin the recovery effort to come out of the safe or standby mode, Operations Planning and Science Team (OPST) will start the process of building the “recovery master sequence,” which is usually a truncated version of the master currently onboard.

For example, if the spacecraft entered into either standby or safe mode after the master sequence for any particular week had been executing for 24 h (assuming that the process of diagnosis and determining the appropriate solution may take a full day or longer), the sequence might be rebuilt exactly as before, but with the first three days missing. This would result in a shortened master sequence (four days long instead of seven in this example), which would begin to execute immediately after a specific downlink pass identified by the OET.

The normal processes and procedures for this emergency build remain in place to the extent that the rapid turnaround time scale allows. The OPST rebuilds the recovery sequence. Members of other teams (OET, ISTs, MST, etc.) check the products for problems prior to delivery to MST. After the delivery to the MST, MST constructs and delivers the command and review products. However, the fast recovery has its pros and cons. One of the problems is that during the fast recovery, less time is given to review or double check the work. Under this time pressure, mistakes can be introduced, although none has been to date. The positive thing about fast recovery is that it helps to get back to normal operations and start collecting science.

Another anomalous occurrence to be considered is the instance of a missed DSN pass. Because of the high data volume produced by MIPS, this can be a potentially serious issue. One method for extending a subsequent pass has been developed using a strategy that allows for the replacement of a portion of the sequence onboard. The master sequence calls and spawns numerous other

processes as it runs, including shorter, relative-timed sequences (generally referred to as slave sequences), which in turn call some fraction of the hundreds of science and calibration requests that are scheduled in that week. It is possible to construct a shortened version of a slave that contains a call to the downlink stop block that has been moved later in time, to extend the downlink pass in question. The replacement slave sequence is then uplinked to the spacecraft and the original version is deleted, before the current master sequence has issued the call or spawn for that particular slave sequence.

III. Conclusion

Spitzer's requirement for adaptable operations and for maximized science viewing efficiency drove an optimization of spacecraft flexibility, adaptability, and use of observation time. This optimization was achieved through the implementation of a multi-engine sequencing architecture coupled with non-deterministic spacecraft and science execution times. To accommodate this design approach and produce sequences in a time-efficient manner, the MST employed a tactic of adapting core sequence generation software and developing scripts to automate labor-intensive processing and sequence product-generation routines. The adapted core software and scripts were then employed in a sequence generation process to accommodate each mission phase and strategy. This strategy achieved mission objectives in a timely and cost-efficient manner.

The MST came onto the mission relatively late (approximately 1.5 to 2 years before launch). At that time, the requirement to maximize science viewing efficiency was in place, but a method to handle the operations complexity was not well understood. Much work had to be quickly done to develop the interfaces, core software adaptations, scripts, and processes to develop workable sequences in a timely, efficient, and cost-effective manner.

Although effective in achieving the viewing efficiency, the 12-engine architecture presented challenges in flight software tests. A fair amount of work had to be performed to get the flight software properly functioning with the multi-engine architecture. In addition, the non-deterministic modeling added much complexity to ground operations. To model the spacecraft's non-deterministic slewing, the ground system had to incorporate the same flight software that was used on the spacecraft. Ground operations could be simplified if the project had decided not to perform non-deterministic modeling on the ground and just accept slave truncations with absolute-timed slave sequence halt routines. The IRAC was extremely command intensive. A modeling run for a one-week IRAC sequence could take approximately 15 h just to obtain a modeled review product (i.e., a modeled .pef). Hardware upgrades to Blade dual-processor 2000 computers did shorten model run times, but model run times still could take in excess of 12 h. Sptizer's multi-engine architecture, as well as its requirement for non-deterministic modeling, also complicated the multimission core software (SEQGEN, SLINC, CMD-TCWRAP) adaptation. Extra testing had to be used to ensure that the multimission core software could adequately account for sequences coming from a multi-engine architecture, and that the core software could effectively interact with the slew model in performing non-deterministic slew modeling.

We strongly recommend that MST personnel be applied to projects earlier in the mission phase so that they can help drive the development of tools, processes, and interfaces as part of the standard development cycle.

Acknowledgments

The work described in this chapter was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA.

References

[1] Lock, P., "SIRTF Sequencing Using the VM," Jet Propulsion Lab., JPL Interoffice Memorandum 450:00:00–09, Pasadena, CA, July 14, 2000.

[2] Bliss, D., "Reduce Costs with Multimission Sequencing and a Multimission Operations System," *Proceedings of the 5th International Symposium on Reducing the Cost of Spacecraft Ground Systems and Operations*, 2006.

Chapter 27

COSMO-SkyMed: Earth Observation Italian Constellation for Risk Management and Security

Francesco Caltagirone* and Arnaldo Capuzi†
Italian Space Agency, Rome, Italy
Roberto Leonardi*
Italian Ministry of Defense, Rome, Italy
and
Sandro Fagioli,‡ Giuseppe Angino,§ and Fabrizio Impagnatiello¶
Thales Alenia Space Italia, Rome, Italy

I. Introduction

COSMO-SKYMED (constellation of small satellites for Mediterranean basin observation) is the largest Italian investment in space systems for Earth observation, commissioned and funded by the Italian Space Agency (ASI) and Italian Ministry of Defense (MOD). The system, currently in the production phase, consists of a constellation of low-Earth-orbit midsized satellites, each carrying a multimode high-resolution synthetic aperture radar (SAR) instrument operating at X-band and a full featured global ground segment to properly exploit space capabilities.

The primary COSMO-SkyMed system mission objective is the provision of services able to quickly answer user needs in the following domains: 1) land monitoring for territory risk management; 2) territory strategic surveillance for intelligence and homeland security; 3) specific defense purposes; 4) management of environmental resources, maritime and shoreline control, and law enforcement;

---

*COSMO-SkyMed Program Manager.
†COSMO-SkyMed Technical Responsible.
‡COSMO-SkyMed Program Director.
§Remote Sensing Engineering Director.
¶COSMO-SkyMed System & Program Manager.

5) topography; 6) scientific applications of institutional entities and academics; and 7) commercial organizations.

COSMO-SkyMed has been conceived to cope with dual-use needs, international partnerships, and integration of the system itself into a multimission framework of cooperating multisensor systems. COSMO-SkyMed is therefore a highly innovative systems, presenting cutting-edge multimission and multi-sensor capabilities implemented through its interoperability, expandability, and multimission design features (IEM) and operational practices.

The benefits of such capabilities are shown, in terms of both costs and user-friendliness, through actual cases that ASI is currently carrying on in international cooperation programs, such as Optical and Radar Federation for Earth Observation (ORFEO) with the French government, in which Pleiades is the optical component of the system; and Italian-Argentinean Satellite System for Emergency Management (SIASGE) with the Argentinean government, where SAOCOM provides complementary L-band SAR capabilities.

COSMO-SkyMed, thanks to its dual capability, will directly either cooperate or compete with other relevant defense systems such as Helios II (France) and SAR-LUPE (Germany) and civilian systems such as TERRASAR and RADARSAT.

The need to achieve COSMO-SkyMed objectives according to a balanced timeline, with respect to similar applications contemporaneously carried out by other international entities, has forced a heavy schedule on the national program.

COSMO-SkyMed has been conceived with the aim to establish a global service supplying a variety of products and services suited to satisfy almost all user/application requirements and most of the potential market demand.

II. System Goals and Key Points

COSMO-SkyMed is designed for observation of areas and targets in all-weather status and illumination conditions (night/day, clear/cloud/rain operations), and to provide end-user services to all registered entities, both civilian and defense, according to their own entitled profile (class, need-to-know, citizen, reference UGS site, etc.).

The constellation of four low-Earth-orbit satellites, sketched in Fig. 1, fulfills high-demanding requirements expressed by ASI and Italian MOD in terms of fast response time, security rules, data confidentiality and type, and quality and number of images per orbit and per day. Thanks to versatility of the SAR instrument, the true enabling core of the system, COSMO-SkyMed can generate image products with different resolutions and sizes, spanning from narrow-field/high-resolution images, through very huge field and mid-low-resolution images, achieving up to 450 images per satellite per day.

Both space and ground segments are conceived with an extendable approach. The space segment, formed by four identical satellites, will progressively achieve full constellation performance, but full Earth coverage and nominal image quality performance will be granted since the first operational satellite. The ground segment will be fully operational before the launch of the first satellite and is featured to satisfy civilian and defense exigencies, as formally expressed by Italian government in 2001. Such a need provides all the functions, infrastructures, and capabilities necessary to operate in accordance with the dual-use specific needs and constraints.

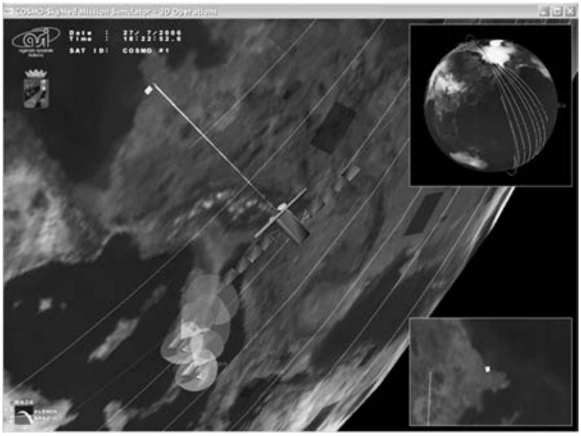


Fig. 1 COSMO-SkyMed operating over Europe.

The mission itself is conceived to allow a substantial growth capability in terms of resources, capabilities, and further applications. The capability to support current and future international cooperations is based on the following system key points: 1) the overall capacity of the fully deployed constellation; 2) the full space segment resource planning capability; and 3) the IEM concepts implemented within ground segment elements, and especially within the user ground segments (GSs).

With a view to properly set up and tune the system, special care has been put on the COSMO-SkyMed Mission Simulator, which has been developed ad hoc by the system prime contractor, Thales Alenia Space Italia S.P.A. (TAS), to give an accurate representation of both space and ground segment components and associated resources. A particular fidelity has been requested on the users definition within the system in terms of engaging rules, priority, quota, and denials.

III. Mission Objectives and Constraints

A. User Needs

Users that can benefit from space remote sensing techniques have been identified in early program phases (1997–1999) together with the relevant requirements on image product characteristics. It is worth noting that, when program duality has been applied on the system definition and development, the most important needs have been related to national security, defense, and risk management applications. Space resources can provide a significant contribution to improve tactical and strategic intelligence capability, to minimize the vulnerability to disasters, etc.

Operating at the same time with civilian and defense users, the commercial applications will fill, and hopefully saturate, the system imaging capability, one of the largest among similar systems.

B. Dual-Use Operations

As mentioned, the system is able to support a true dual-use scenario that foresees various defense and civilian user classes, in both national and international contexts. In other words, the system will be able to provide the required/agreed level of service to each user, asking for the capability to be configured in a flexible and expandable way for supporting new users. The system overall coordination function shall be centralized (and based on priority rules). To cope with this exigency, enabled in Italy by specific national regulation, a relevant security know-how has been developed by all entities (ASI, National Security Agency (ANS), and industry) to implement and put in place a system that really meets user needs from one side and specific employment and security rules from the other one. The cumulated experience in this field is unique and has been recognized by various international entities potentially interested in similar applications [Global Monitoring for Environment and Security (GMES), Galileo, Multinational Space-based Imaging System (MUSIS) for surveillance, reconnaissance, and observation].

The final acceptance for security purposes will be managed directly by the Italian ANS.

C. Cooperation Scenario

Since its very beginning, the COSMO-SkyMed program has been conceived to be implemented within an international scenario covering both development of the infrastructures and utilization of the system. According to this policy, ASI started to establish several bilateral formal contacts to identify and define the level of interest and potential involvement of other national space agencies/authorities. In particular, within the ORFEO context, COSMO-SkyMed was recognized to be the “radar component” of the Italian-French binational system.

Even if no formal agreement exists, the planned Sentinel-1 in the frame of the GMES program has been defined on the basis of COSMO-SkyMed design guidelines. The nature of the system was also to ensure the continuity of service that is now planned for at least 15 years, starting from the launch of the first satellite.

The system is thus a direct candidate for the European Earth observation needs of the European Union and a precursor system for GMES.

IV. System Architecture

COSMO-SkyMed is then a multisatellite Earth observation system based on a sensor capable of acquiring radar backscatter information at 9.6 GHz. The fulfillment of previously mentioned dual-use applications calls for the following general performance characteristics:

1) Full global coverage (say “imaging everywhere”), achieved with the selected center frequency at X-band and orbit/phase selection on the basis of achievable swath widths in the compatible incidence angle range.

2) Fast response time (from the data/service user request up to the data/service delivery to that requiring user), achieved with a proper international ground station scenario identification, in which cooperation politics has given a relevant contribution.

3) Very good image quality to allow a robust image interpretability at the requested scale of analysis, achieved with an innovative SAR instrument digital design and specialized SAR processors within user ground segment (UGS), covering all levels from L0 to L1d and including a certain number of L2 value added applications.

4) Collection of large areas with single-pass operation, granting along-track phase coherence that enables interferometric applications (topography), achieved with both flexible SAR instrument timing and SAR antenna performance and reprogramming capability together with mosaic processors in the processing facilities.

5) Acquisition of a sufficiently large and interpretable image in a single pass, thanks to a significant space-to-ground data download rate.

6) Acquisition of a homogeneous and comparable multitemporal data set, characterized by adequate spatial and spectral resolution suitable to perform analyses at different scales of detail, obtained with highly controlled frozen orbit, specific calibration data for each satellite, and advanced mission planning and system programming features.

A. Space Segment

The COSMO-SkyMed space segment is composed of a constellation of four SAR satellites, as sketched in Fig. 2.

The satellite, equipped with a SAR instrument engaging several hundreds of MHz bandwidth at X-band, is fully three axes stabilized and optimized for the COSMO-SkyMed nominal 237/16 sun-synchronous orbit whose altitude is in the range 620–652 km. Figure 3 gives an idea of the SAR satellite shape.

The main satellite design characteristics, apart from the SAR instrument-specific features whose operation modes are synthesized in Fig. 4, are 1) direct current (DC) power availability in terms of peak and average levels; 2) satellite command ability robustness [satellite protection, access rules, failure detection and instant

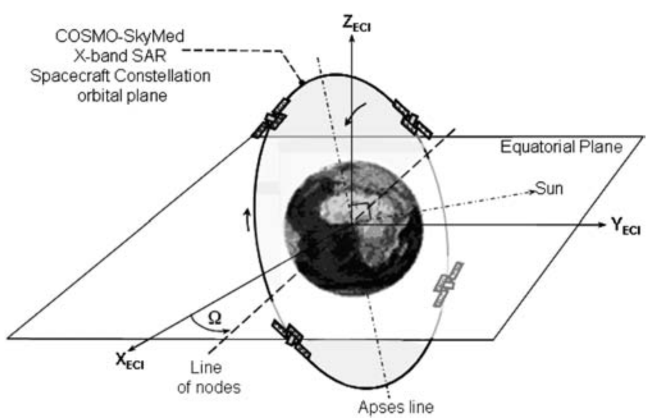


Fig. 2 COSMO-SkyMed space segment.

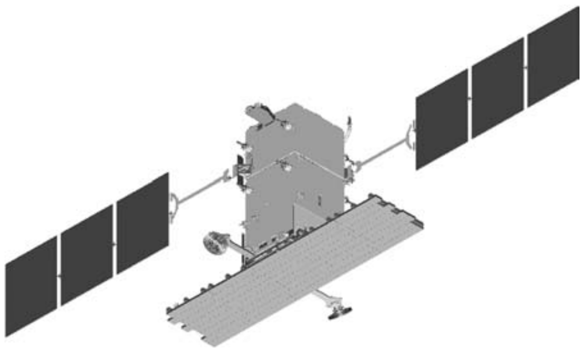


Fig. 3 SAR satellite of COSMO-SkyMed constellation.

recovery (FDIR) for autonomous failure detection and safe management]; 3) pointing and agility performance to ensure both right-looking and left-looking SAR imaging; 4) time-to-position management to correctly implement spotlight imaging operations; and 5) navigation and autonomy operation capability.

The satellite is designed to autonomously manage its mission at least in the next 24-h horizon. The command plan uploaded from available ground stations details all operations to be carried out on the satellite according to a time-tagged ordered sequence.

An aggressive system mission profile is supported by navigation, pointing, and attitude charged subsystems. Challenging real-time navigation is guaranteed by an onboard global positioning system (GPS) receiver and precise orbit determination software (SW) running within the integrated control system (ICS).

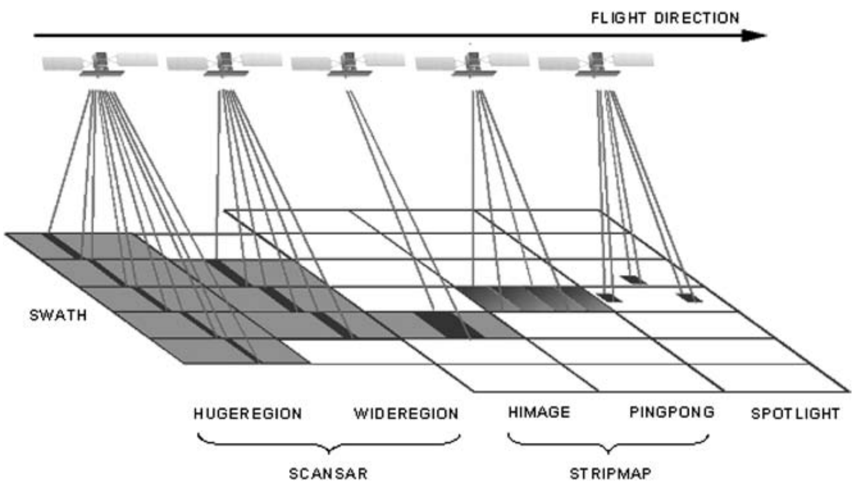


Fig. 4 SAR instrument operation modes.

Extreme pointing performance is guaranteed by a novel design of the body top floor direct integrating star trackers close to the SAR antenna mounting staffs. Finally, the attitude control system is agile enough to implement in a few minutes the roll maneuver necessary to switch from nominal right-looking imaging mode to the left-looking mode raised under exceptional circumstances when the COSMO-SkyMed very urgent system mode is declared. This feature is fundamental to grant the revisit time customer requirements.

B. Global GS and IEM Capabilities

The proposed high-level architecture is based on an element-based approach, identifying 1) the mission planning and control center (CPCM), 2) the satellite control center (CCS), 3) the user ground segments (I-CUGS and I-DUGS), and 4) the support elements (SSE).

All satellites are commanded, controlled, and monitored by the CCS, which also ensures flight dynamics and telemetry & telecommand (TTC) control.

The devoted CPCM has been identified and realized in terms of functions, operational rules, and performance.

Finally, the system exploitation is obtained within the two sections of user ground segments (UGS), one for civilian and the other for military applications. Mobile UGS completes the acquisition/exploitation chains as follows:

1) The civil UGS and its primary acquisition station are located in the south of Italy.

2) The military UGS is located in the center of Italy.

3) The mobile acquisition and processing stations (MAPS) are mobile stations to be deployed on crisis theatres to allow the immediate availability of the data.

4) One polar station at Kiruna together with a devoted station placed at Cordoba that sustains and is part of the SIASGE cooperation program between Italy and Argentina.

The multimission objective is pursued by ASI and the Italian MOD through a two-fold strategy, i.e., by means of 1) establishing agreements with international partners for sharing the Earth-observation (EO) assets, and 2) driving the COSMO-SkyMed system design and interfaces to achieve such an objective.

To fulfill these requirements, the COSMO-SkyMed architecture relies on the following design key points, whose applicability joint domains are sketched in Fig. 5:

1) Interoperability: Interoperability is the capability of exchanging data and information with external heterogeneous systems according to pre-defined agreed modalities and standards, and irrespective of internal design of the cooperating parts. The COSMO-SkyMed architecture implements the standard catalog interoperability protocol based on the Committee on Earth Observation Satellites (CEOS) guidelines, through which it provides access to a variety of EO systems worldwide, to cover the observation needs of the largest number and typologies of users, mainly for civilian institutional, commercial, and scientific purposes.

2) Multimission, Multisensor: Multisensorality is the system ability to request, process, and manage data related to different observation sensors. For the timebeing, COSMO-SkyMed is envisaged to manage the following sensor data: X-band SAR data from its own spacecraft constellation, SAR Bistatic data for

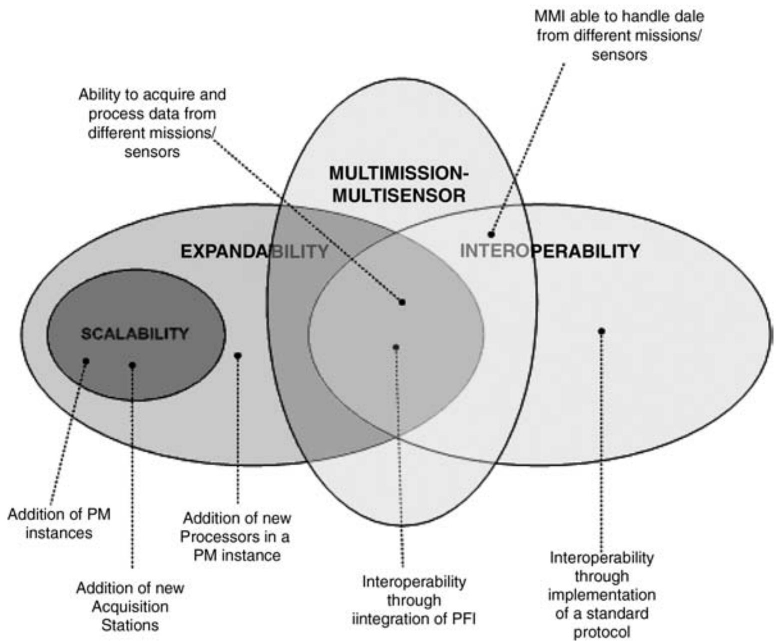


Fig. 5 COSMO-SkyMed IEM concepts implementation.

interferometry, L-band SAR data from Argentinian SAOCOM satellites, and optical imaging data from the French Pleiades constellation. Multisensorality concerns a number of end-to-end functional chains, in turn: 1) mission programming chain, including depositing, analysis, harmonization of programming requests for acquisition from multiple sensors, up to the scheduling of the related image data takes, 2) image chain, to process data generated from different sensors, then to extract and to correlate the imaging features, and 3) user service chain providing end-users with the capability of searching and ordering products from different sensors, as well as multisensor (e.g., coregistration) products. In the same way, the multimission capability represents the ability of the system to manage, provide, acquire, and exchange information with different EO missions.

3) Expandability: Expandability is the ability of an architecture to embody *mission-specific* components "imported" from a partner's EO system, thus designated as a partner's furnished items (PFI). The COSMO-SkyMed architecture is designed to manage several PFIs from different partners' systems, such as 1) acquisition chain, 2) processing chain, and 3) programming chain, in order to *locally* achieve multimission and multisensor capabilities. Reciprocally, COSMO mission-specific components can be configured as PFI to be exported toward a partner's EO systems. A clear specification of PFI's scope and interfaces constitute key issues for COSMO architecture expandability feature.

A technically sound and cost-effective expandability shall avoid unnecessary duplication of components for implementing functional chains related to different

missions or sensors. This mainly depends on two aspects: 1) a scalable GS architecture, and 2) a clear identification of mission-generic elements, interacting with mission-specific elements through given interfaces.

COSMO-SkyMed GS achieves both aspects. Scalability—the capability to fulfill increasing performance needs (e.g., a higher number of products) by adding "copies" of modules already part of the system architecture, and properly configuring them to make the whole system working at the requested performance levels—is ensured through the ground segment modular design, which allows the increase of capacities through plugging-in modules easily configurable and managed by the pre-existing architectural infrastructure.

This modularity also helps in identifying and implementing mission-generic and mission-specific modules. The latter are conceived as plug-ins into the common infrastructure provided by the mission-generic components. For example, the processor encapsulator (PE) is a UGS mission-generic element that provides services to a variety of (mission-specific) instrument processors.

C. System Validation and OVT Outdoor Validation Test

Because COSMO-SkyMed is one of the most complex EO systems that embarks one of the most advanced spaceborne synthetic aperture radar ever conceived, both customer and prime contractor agreed to give a direct demonstration of SAR imaging operation. This campaign has been named the Outdoor Validation Test (OVT) and has been managed directly at system level.

Confident of both spacecraft and ground segment, already designed and developed by main industrial suppliers (TAS and Telespazio), the SAR instrument is really new, both in terms of operation complexity and in type and number of active transmitter/receiver (T/R) modules at antenna level.

Figure 6 shows a portion of the flight SAR antenna (half of central panel) mounted and made operational at Alcatel Alenia Space Italia premises.

Fig. 6 Flight SAR antenna portion used for Outdoor Validation Test.

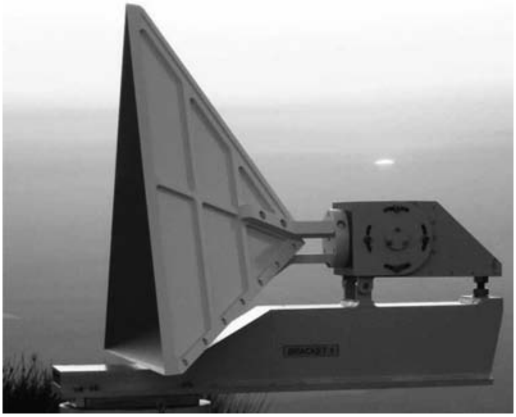


Fig. 7 OVT calibrated "target."

Using special calibrated corner reflectors (see Fig. 7) several tens of kilometers far from the SAR instrument, completed by the true flight electronics, the SAR payload was engaged with an I-SAR special mode.

The test demonstrated full matching with expected theoretical performance thanks to the very advanced pulse shape control exercised both at amplitude and phase level on digital synthesized waveforms.

The test has shown full adherence with expected resolutions and noise thresholds in both high, mid-, and low-resolution modes. Furthermore, the resulting data set, even if not representative of a SAR instrument flying on the nominal orbit at 7 km/s, will be used for a preflight SAR processor validation within the UGS instances.

V. Performance

The system guarantees a worldwide access capability, starting from the first deployed satellite.

The imaging system service is mainly characterized by two time performance figures: the revisit time and the system response time. The latter is the sum of the programming delay, the access delay, and the data aging contributions.

Revisit time defines the time interval between two successive imagings of the same area or target of interest. It depends on constellation design and matches customer requirements worldwide.

Response time is the overall delay between the depositing of a user request and the delivery of the required product to that user. This depends on both constellation design and the position of selected ground stations. The S-band TTC and X-band receiving ground station placement over Italian territory matches requirements over the European area of interest. Small response time performance is achieved using a polar TTC station.

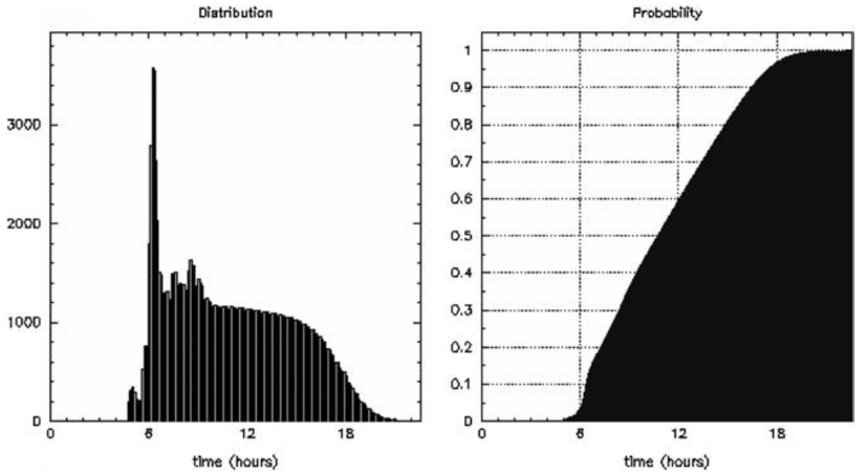


Fig. 8 Full constellation: response time statistics.

Figure 8 shows achievable response time statistics by considering all the contributions that add to the end-to-end performance.

VI. Defense and Homeland Security

Concerning defense and homeland security, COSMO-SkyMed envisages the following applications:

1) Surveillance of territories, national borders, and shorelines.

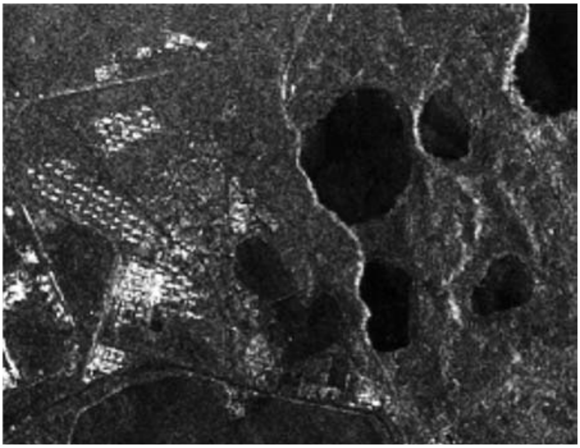


Fig. 9 Security image exploitation example.

2) Intelligence applications, target detection, evaluation of activities, and temporal change detection.

3) Characterization and monitoring of crisis area, damage estimation, and operations assessment.

4) Monitoring of vessel traffic, e.g., for clandestine immigration.

5) Image system integrated with command and control system for decision support.

Strategic and tactical applications need a combination of intelligence and surveillance imagery, together with real-time communications and information processing technologies. In particular, the imagery intelligence (IMINT) aims to collect data about the operative scenarios. COSMO-SkyMed SAR imagery will extend the IMINT objectives.

In fact, all-weather day-and-night observations from space can provide the capability of finding and discovering the presence of targets (detection), of naming an object by class (identification), of describing configuration and details (description), of examining the target to learn what the target is made of (analysis).

As for the system requirements involved by military applications, the different phases of the interpretation process previously mentioned need different spatial resolutions. To reveal the presence of sensible targets on a wide area, the detection phase requires wide swath and medium resolution, similar to those ones provided in the example given in Fig. 9. The other phases of the interpretation process need higher resolution and smaller swath dimensions, as their end is the investigation of the features relative to a previously detected target.

VII. Conclusion

COSMO-SkyMed is a primary example of a dual-use Earth-observation system, cooperating and interoperating with a Partner's EO systems with the aim to provide multimission integrated services to the largest number and variety of end-users worldwide starting from the beginning of 2007, when the system will start its operational phase. Its design characteristics are deeply rooted in the "native" versatility of the key elements composing the COSMO-SkyMed system: the SAR spacecraft constellation (four satellites equipped with multimode high-resolution SAR instruments, and the dual ground segment.

This versatility is also fundamental to optimally provide the appropriate services that cover different utilization needs of defense and civilian domains, providing for images at different sizes, and resolutions, with different geolocation accuracy and priority. This schema can be adopted also in the multimission cooperation scenario, tuned to international defense cooperation and worldwide civilian use.

These features designate COSMO-SkyMed as a system capable of providing "Institutional Awareness," to make proper decisions to prevent and manage a worldwide crisis, covering also key mission requirements for global environmental monitoring and citizen security, whose international agencies and organizations are currently being defined. This characterizes COSMO-SkyMed as the best Italian contribution to such an enterprise, a precursor system and a cornerstone that makes the concepts of the future feasible today.

Moreover, the program is providing the true opportunity of implementing advanced technological research funded and driven in accordance with national

development strategies for dual technologies. All previous investments in the last two decades gave satisfactory returns, providing products and methodologies able to push Italian institutional and industrial assets in a dominant position.

Finally, the Italian Space Agency and the Italian MOD are acting a true and tangible spin-off with respect to civilian institutions and defense operators. Thus the intrinsic characteristics of developed applications are generating social benefits on a large scale and will (hopefully) induce a new commercial deal.

Acknowledgments

The authors would like to acknowledge the work performed during the COSMO-SkyMed C/D Phase contract by all the other members of the industrial team and of the ASI and Italian MOD project and support team members who are not explicitly mentioned in this chapter.

This page intentionally left blank

VIII. Moon, Mars, and Beyond

This page intentionally left blank

Chapter 28

SMART-1 Lunar Mission: Operations Close to Moon Impact

Jurriaan de Bruin*
VEGA, Darmstadt, Germany
Octavio Camino-Ramos† and Johannes Schoenmaekers‡
European Space Agency, Darmstadt, Germany
and
Per Bodin§
Swedish Space Corporation, Solna, Sweden

Nomenclature

ΔV = delta velocity, km/s (unless indicated otherwise)
L_{RW} = reaction wheels angular momentum, N·m·s
$L_{S/C}$ = spacecraft angular momentum, N·m·s
V = velocity, km/s

I. Introduction

SMART-1 is the first of the ESA's Small Missions for Advanced Research in Technology. It demonstrated orbit raising from geostationary transfer orbit to the moon using solar-electric propulsion. In November 2004 SMART-1 successfully maneuvered into moon orbit. Since January 2005 SMART-1 has been in its operational orbit performing scientific operations that were interrupted only by a

*Spacecraft Operations Engineer.
†Spacecraft Operations Manager, European Space Operations Centre.
‡Flight Dynamics Manager, European Space Operations Centre.
§Aerospace Engineer.

one-month reboost phase in September 2005 to re-optimize the orbit. Toward the end of mission, a series of orbit control maneuvers was done to influence the impact date and location so that Earth-based telescopes could observe the impact. On 3 September 2006 at 05:42:22 Coordinated Universal Time (UTC), SMART-1 impacted the moon in an area called the Lake of Excellence. The impact was observed and confirmed from Earth by radio and infra-red telescopes.

A. Spacecraft

SMART-1 is a three-axis stabilized spacecraft consisting of a 1-m^3 central box and two solar array wings. The complete spacecraft weighed 370 kg at launch. The central structure is designed around a xenon fuel tank with a capacity of 49 liters, containing 82.5 kg xenon at launch. A central equipment deck contains most spacecraft units, with the exception of high heat dissipaters. The solar arrays are sized to deliver 1850 W at the beginning of life. Split into two wings of three panels each, the solar arrays span 14 m tip to tip. The solar arrays are positioned on opposite sides of the spacecraft and are able to rotate. In the orbit-raising phase, this allows the thrust vector and solar arrays to be optimally pointed at the same time. Batteries provide power through eclipse phases of the mission, and they are sized to support a maximum eclipse length of 2.1 h (no thrusting). Primary propulsion is performed by the solar-electric propulsion thruster. The electric propulsion system is equipped with a software-controlled mechanism to change azimuth and elevation of the thrust vector. Attitude control is performed by reaction wheels, with hydrazine thrusters used in lower spacecraft modes and to perform reaction wheel offloading. Attitude information is obtained through a combination of sun sensors, gyros, and startrackers. The data-handling subsystem contains cold redundancy, with autonomous failure, detection, and isolation (FDIR) software handling any single failures. The onboard software is designed with a high level of autonomy. Normal operation can continue in an absence of ground contact for 10 days, and the spacecraft can survive in safe mode for a period of two months or more [1].

B. Mission

In summary, the mission phases and trajectory details are as follows:

1) Launch and early orbit phase. Launch on 27 September 2003, initial orbit $7029 \times 42{,}263$ km.

2) Van Allen Belt escape. Continuous thrust strategy to quickly raise the perigee radius; escape phase completed by 22 December 2003, orbit $20{,}000 \times 63{,}427$ km.

3) Earth escape cruise. Thrust around perigee only to raise the apogee radius.

4) Moon resonances and capture. Trajectory assists by means of moon resonances; moon capture on 11 November 2004 at 310,000 km from the Earth and 90,000 km from the moon.

5) Lunar descent. Thrust used to lower the orbit, operational orbit 2200×4600 km.

6) Lunar science. Until the end of lifetime around September 2006, interrupted by a one-month reboost phase in September 2005 to optimize the lunar orbit and

a second reboost phase in June/July 2006 to fine tune the impact date and location.

7) Moon impact on 3 September 2006 at 05:42:22 UTC, in an area called the Lake of Excellence (34.4°S, 46.2°W).

C. Payload

SMART-1 is testing not only solar electric propulsion but also other deep-space technologies and will perform scientific observations of the moon. Among other investigations, it investigates the origin of the moon and searches for ice in the craters at the moon's South Pole. Experiments onboard SMART-1 are the following:

1) Electric Propulsion Diagnostic Package (EPDP) to monitor the working of the propulsion system and its effects on the spacecraft.

2) Spacecraft Potential, Electron and Dust Experiment (SPEDE) to also monitor the effect of the propulsion system and to investigate the electrical environment of the Earth-moon space.

3) X/Ka-band Telemetry and Telecommand Experiment (KaTE) to test more efficient communication techniques with Earth.

4) Radio Science Investigation with SMART-1 (RSIS) uses the KaTE and Advanced Moon Imaging Experiment (AMIE) instruments to investigate the way the moon wobbles.

5) Onboard Autonomous Navigation (OBAN) is a software to allow the spacecraft to guide itself to the moon.

6) Advanced Moon Imaging Experiment (AMIE) to test a miniaturized camera and take color images of the moon surface.

7) SMART-1 Infrared spectrometer (SIR) to search for ice and make a mineralogical mapping of the moon.

8) Demonstration of a Compact Imaging X-ray Spectrometer (D-CIXS) to investigate the composition of the moon.

9) X-ray Solar Monitor (XSM) to calibrate the D-CIXS data and study solar x-ray emission.

D. Ground Segment

SMART-1 is operated from the European Space Operations Centre (ESOC) in Darmstadt, Germany. This location occupies around 600 people (about two-thirds contractors) and is responsible for operating most of ESA's scientific missions. In addition, it holds the development responsibilities for all ESA ground stations, and associated networks in cooperation with international partners, ESA mission analysis for future missions, and all flight dynamics operational services. The center is the home of the Spacecraft Operations Control System (SCOS) used to monitor and control spacecraft in multiple control centers around the world.

II. Nominal Science Mission

In November 2004 SMART-1 successfully maneuvered into moon orbit, a major milestone in the mission. SMART-1 made history with several notable firsts, including being the first mission to escape Earth orbit with the use of electric

propulsion, the first to use electric propulsion to enter into orbit around another celestial body, and being the first European lunar mission [2].

Shortly after capture the approximately three-month lunar decent phase began. The objective of the lunar descent phase was to adjust the initial lunar capture orbit to its operational one. All lunar operational orbits that were considered for the SMART-1 science phase were polar and had an initial argument of perilune that let the perilune drift over the South Pole [3]. The perilune drift is retrograde in all cases, starting with an initial argument larger than 270 deg. For this type of lunar orbit, third-body perturbations not only cause an argument of perilune drift but over time also affect the perilune and apolune distance. For an argument of perilune larger than 270 deg, the perilune distance increases and apolune distance decreases over time, and conversely, for an argument of perilune smaller than 270 deg, the perilune distance decreases and apolune distance increases over time. For the SMART-1 orbital evolution, it means that the eccentricity decreases initially and then increases again until the perilune distance is smaller than the moon radius and the spacecraft impacts the moon. The rate of change for the eccentricity strongly depends on the initial value of the argument of perilune. The choice of the initial apolune altitude determines the fuel consumption and also has a strong impact on the scientific merit. Orbits with a low apolune are more costly to reach in terms of propellant and the target orbit acquisition takes longer. However, they also offer superior coverage quality.

SMART-1 arrived in moon orbit with more xenon propellant remaining than previously assumed in the prelaunch mission design. In-flight experience learned that the solar array power and the efficiency of the solar electric propulsion system exceeded the assumptions made prior to launch. These assumptions were known to be conservative. It was decided that the extra xenon was to be used for optimizing the operational orbit for maximum science return and extending the mission lifetime. To maximize science return, it was decided to lower the initial apolune distance from 10,000 km (as per baseline orbit) to approximately 4600 km initially. To extend the mission lifetime, a reboost phase was needed after approximately six months of science operations to correct for the argument of perilune drift. By the end of February 2005, the spacecraft reached the newly defined operational orbit of 2200×4600 km (perilune $\times$ apolune distance). Note the difference with the baseline 2000×10000 km orbit. The nominal science mission lasted six months from the time that the spacecraft reached its operational orbit in February 2005.

Table 1 shows the orbital elements, apolune/perilune distance, and the xenon consumption for the entire moon phase. Figure 1 illustrates the spacecraft-to-moon-center distance from capture to impact. The phases that used propulsion of any kind to change the orbit are indicated in the figure.

III. Extended Science Mission

On 10 February 2005 ESA’s Science Program Committee endorsed unanimously the proposed one-year extension of SMART-1, pushing back the mission end date from autumn 2005 to autumn 2006. To extend the mission, the last remaining propellant for the electric propulsion was to be used for an orbit reboost. This reboost phase took place in August/September 2005 and made the electric

Table 1 Orbital evolution and xenon consumption for the entire moon phase

EPOCH	2004/11/15 17:47:38	2005/02/28 05:18:39	2005/08/01 06:42:55	2005/09/19 08:32:45	2006/06/19 06:46:53	2006/07/03 07:47:16	2006/09/03 05:42:21
Phase description	Capture	End descent	Start 1st reboost	End 1st reboost	Start 2nd reboost	End 2nd reboost	Impact
Consumed Xenon (%)	71.5	87.9	87.9	99.7	99.7	99.7	99.7
Consumed Hydrazine (%)	3.7	6.6	10.8	13.3	25.2	50.8	54.3
Pericentre distance (km)	6704.3	2208.7	2170.4	2189.7	2016.1	2034.7	1737.2
Apocentre distance (km)	53207.6	4618.2	4634.8	4644.0	4821.3	4889.7	5191.3
Semi-major axis (km)	29956.0	3413.4	3402.6	3416.9	3418.7	3462.2	3464.3
Eccentricity	0.776	0.353	0.362	0.359	0.410	0.412	0.499
Inclination (deg)	81.1	90.1	90.2	90.2	90.8	90.8	91.2
Ascending node (deg)	246.5	236.5	237.4	237.7	239.4	239.6	240.1
Argument of Pericentre (deg)	308.0	286.2	243.8	292.9	226.3	223.9	216.4
True anomaly (deg)	0.0	180.0	180.0	180.0	180.0	180.0	357.9
Osc. orbital period (hours)	129.2	5.0	4.9	5.0	5.0	5.1	5.1

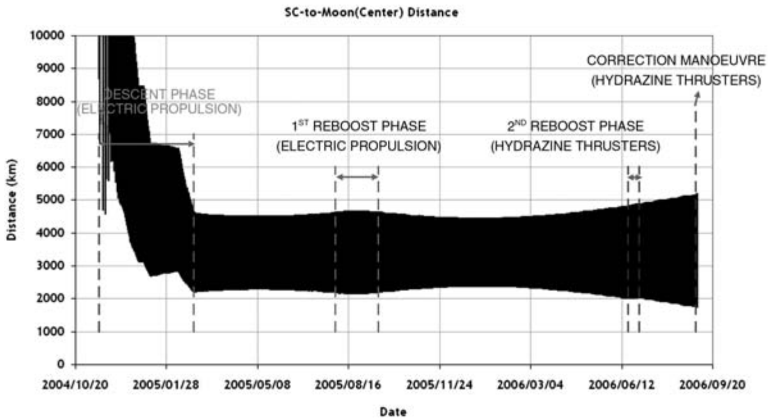


Fig. 1 Spacecraft-to-moon-center distance from capture to impact. Phases that used propulsion of any kind are indicated.

propulsion system consume the remaining xenon up to 99.7% of the initial xenon available. The objective of the first reboost phase was to correct for the argument of perilune drift and thereby extend the mission lifetime but also increase science opportunities around the lunar South Pole. The orbit after the reboost phase resembled in many ways the orbit just after the lunar descent phase and has extended the mission lifetime by another nine months, plus three months with a perilune altitude below 300 km. Table 1 shows that at the start of the reboost phase the argument of perilune had drifted ~40 deg from the initial 286.2–243.8 deg. At the end of the reboost campaign, the argument of perilune was corrected to 292.9 deg. Table 2 summarizes the achievements of the electric propulsion system.

IV. Moon Effect on Spacecraft Subsystems

Toward the end of mission, the different spacecraft subsystems started to be affected by the close proximity of the moon around perilune.

Table 2 Electric propulsion system in numbers

Number of pulses	844
Total number of hours fired	4958.3
First pulse	2003/09/30 12:25
Last pulse	2005/09/17 18:45
Number of valve activations	1,256,505
Initial xenon mass (kg)	82.5
End-of-mission Xenon mass (kg)	0.280
Useable xenon (kg) of the remaining amount	~0.060
Power set range (W) used during the mission	649 – 1417
Number of flame-outs	38

A. Thermal

The Thermal Control and Structures Division at the ESA European Space Research and Technology Centre (ESTEC) studied the influence of the moon on the thermal balance of the spacecraft during the final mission phase [4]. The solar arrays were identified as the main area of concern, as they are exposed to radiation directly and have a low thermal inertia and a high external emittance. The worst case was to be expected in May 2006. At that time the sun direction coincided with, or was close to, the orbital plane and perilune was on the illuminated side of the moon (see Fig. 2). For some time every orbit the spacecraft was in between the sun and the moon at relatively close distance to the moon. In that situation the front side of the solar arrays gets illuminated by the sun and the back side is exposed to a fully illuminated moon (and thus receiving full albedo and infrared radiation input). Maximum solar panel temperature to be reached during that period was predicted to be below the qualification temperature but getting very close to it. As per requirements, the solar panels are qualified up to 120°C and the individual components up to higher temperatures than that even.

With the support of project and industry, the Flight Control Team at ESOC decided to take no operational risk and developed procedures to offpoint the solar panels to avoid those high temperatures. In May/June 2006 the SMART-1 Flight Control Team commanded a 35 deg solar panel offpointing, resulting in an immediate drop of the solar panel temperatures of about 10°C. As a side effect, the solar panel offpointing reduced also the power generated by the solar arrays by ~18%. This reduction in solar array output power was never a problem, as the solar arrays are sized to power the electric propulsion engine, and the engine was not going to be used at that time.

B. Pointing Constraints

SMART-1 followed different types of attitude pointings around the moon to meet the scientific objectives of the mission without violating platform and payload constraints and to ensure periodic Earth communication with the medium gain antenna. Closer to the moon it became more difficult not to violate platform constraints.

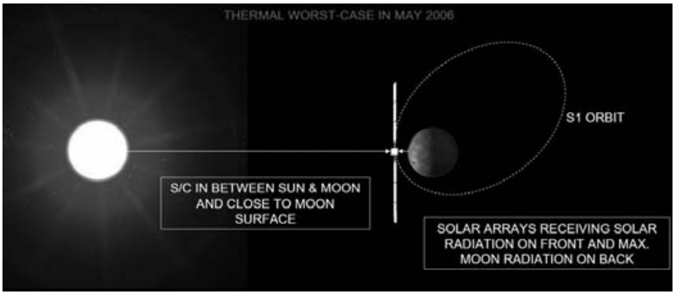


Fig. 2 Sun, Earth, spacecraft configuration during the thermal worst-case period in May 2006.

The Swedish Space Corporation (SMART-1 prime contractor) was asked to analyze the feasibility of the different spacecraft attitude pointings at altitudes around the moon below 300 km [5]. The analysis done by industry showed the following:

1) True moon spot pointing was no longer feasible at altitudes below 240 km because of the high spacecraft rates that it would require [5]. True moon spot pointing is defined by the spacecraft *z*-axis tracking a target of scientific interest on the moon surface for a certain period of time (e.g., in the order of 0–60 s). The spacecraft *y*-axis is aligned perpendicular to the sun nominally, but during push-broom operations this constraint can be relaxed. Below 240 km the required spacecraft rate is higher than 0.45 deg/s, which is the spacecraft rate for safe mode entry.

2) Nadir pointing was no longer possible at altitudes below 112 km, because both startracker cameras will have the moon surface in the field of view [5]. Nadir pointing is defined by the spacecraft *z* axis pointing to the center of the moon (means scientific instruments are pointing to the moon surface) and spacecraft *y* axis aligned perpendicular to the sun. To clear one of the two startracker cameras, while sacrificing the other, the spacecraft had to be tilted off-nadir below 112 km altitude (see Fig. 3).

V. Moon Impact

Toward moon impact a number of activities were planned to ensure that the impact was going to be visible from Earth, and that a campaign was organized to observe it.

A. Innovative Reboost Phase

The idea of a reboost phase toward the end of the mission comes from the fact that without orbit control the spacecraft would have impacted the moon on the far side, away from ground contact and visibility. To observe the impact from Earth, it was decided to move the impact date from middle of August to beginning of September 2006, requiring a ΔV of ~11.8 m/s in total and raising the perilune by approximately 90 km. Use of the electric propulsion system was ruled out for this reboost phase, as all propellant reserves were used during the previous reboost

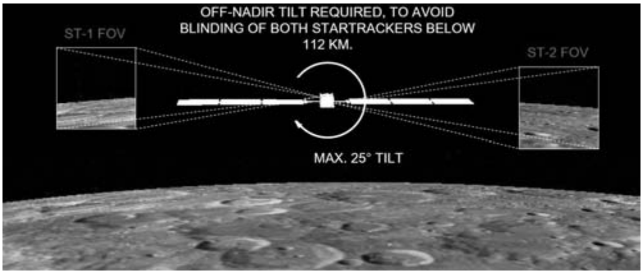


Fig. 3 Nadir pointing is no longer possible below 112 km.

phase. Instead, the hydrazine subsystem was going to be used to do a series of small ΔV maneuvers.

By design, the hydrazine subsystem is optimized for reaction wheel offloadings, not for ΔV maneuvers. The thrusters are mounted to the bottom deck (spacecraft $-z$ panel) using thruster brackets, as illustrated in Fig. 4. The thrust directions of the thrusters are all aligned in one plane, parallel to the spacecraft x–y plane and asymmetric about the spacecraft center of mass, which emphasizes that the hydrazine subsystem is not designed to produce ΔV. Note that when firing all thrusters at the same time, there is no net force on the spacecraft. During a reaction wheel offloading, not all thrusters are firing (at the same time), however, as the offloading aims to change the reaction wheel angular momentum. The asymmetric firing of the thrusters during a reaction wheel offloading produces a small ΔV as a side effect, and this can be used to change the orbit. During the reboost phase it was planned to do a series of reaction wheel offloadings around apolune, transferring the spacecraft angular momentum from one side to the other, and vice versa. In between reaction wheel offloadings, the spacecraft would then be slewed such that the ΔV is aligned with the flight direction. Figure 5 shows part of the reboost strategy for clarification.

The reboost phase started on 19 June 2006 and finished on 2 July 2006. During this period the spacecraft did seven reaction wheel offloadings per orbit, transferring $+/-2.34$ N·ms angular momentum between the spacecraft $+/-y$ axis, in a period of ~3 h centered around apolune. In between reaction wheel offloadings, the spacecraft was performing 180-deg slews around the $+/-Y$ axis of the spacecraft at an angular rate of ~0.22 deg/s. The ΔV maneuvers were a success and achieved the objective of changing the impact date to 3 September.

B. Correction Maneuver

Four days before the planned impact, the Mission Control Team at ESOC decided to act on the latest estimates of the elevation of the moon terrain surrounding the last couple of perilune passages. This information was kindly sent to ESA by Anthony C. Cook from Nottingham University and Mark R. Rosiek from

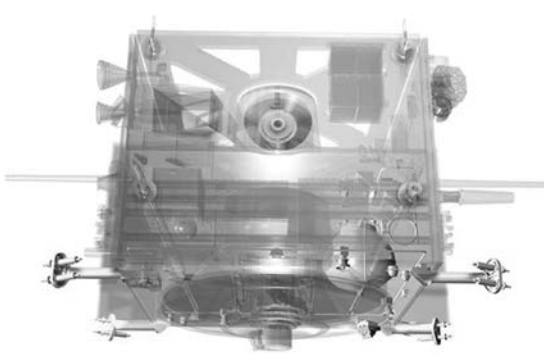


Fig. 4 Thruster arrangement. Each thruster bracket has two thrusters attached to it, the nominal and redundant one. Credits: ESA/Swedish Space Corporation.

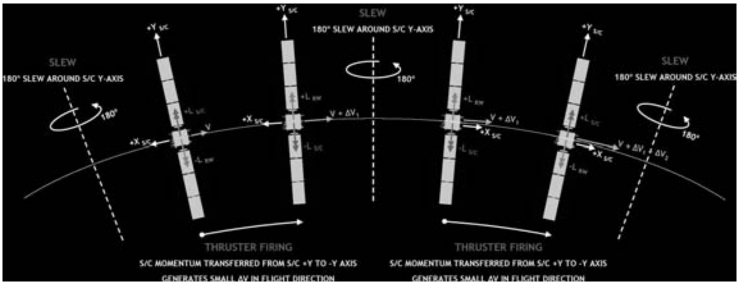


Fig. 5 Part of the reboost strategy.

the United States Geological Survey (USGS) organization some days before that. Nottingham University is specialized in three-dimensional digital image interpretation. Stereo image analysis combined with current topographical models of the moon, as well as information obtained from photos taken by the SMART-1 scientific camera (AMIE) of the potential impact area in midsummer and on 19 August, indicated a high probability that there could be an additional couple of hundred meters of landscape elevation in the orbit before the planned impact. In that case SMART-1 would possibly clip the rim of a medium-sized crater, Clausius, located at 43.5ºW and 36.5ºS. This conclusion was supported by additional data analysis provided by the USGS.

Figure 6 is a three-dimensional plot of the impact area that shows the last couple orbits before and after the planned impact. The spacecraft trajectory (in blue) disappears where the orbital altitude is below the moon surface. It shows that the Clausius crater is a liability.

Based on this new information, all parties agreed that at that stage the first priority was to take whatever measures were necessary to maximize the probability

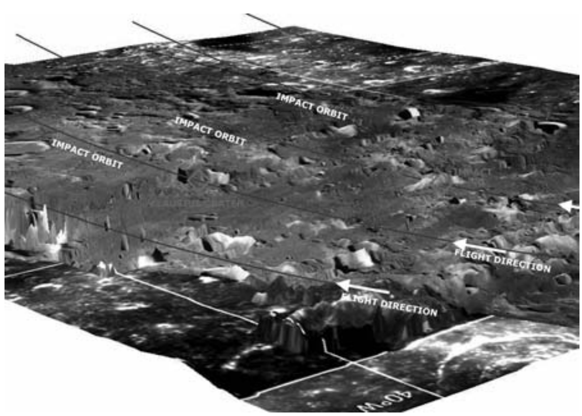


Fig. 6 Three-dimensional plot of the impact area and spacecraft trajectory. Credits: USGS/University of Nottingham.

of impacting as planned on the morning of 3 September around 05:42 UTC. As a result, during the night of 1–2 September, mission controllers conducted correction maneuvers (similar to the ones discussed in the previous section) aiming to boost the height of perilune of the penultimate orbit, while maintaining the intended impact time and location. The correction maneuvers successfully achieved this aim, boosting the perilune by 592 m.

C. Moon Impact

On 3 September, early in the morning at 05:42 UTC, a small flash illuminated the surface of the moon as the ESA's SMART-1 spacecraft impacted onto the lunar surface. SMART-1 scientists, engineers, and space operations experts witnessed the final moments of the spacecraft's life during the night between Saturday 2 and Sunday 3 September at the ESOC, in Darmstadt, Germany. The confirmation of the impact reached ESOC at 05:42:22 UTC, when ESA's New Norcia ground station in Australia suddenly lost radio contact with the spacecraft. SMART-1 ended its journey in the Lake of Excellence, in the point situated at 34.4ºS latitude and 46.2ºW longitude.

The SMART-1 impact took place on the near side of the moon, in a dark area just near the terminator (the line separating the day side from the night side), at a grazing angle between 5 and 10 deg and a speed of ~2 km/s. The impact time and location were planned to favor observations of the impact event from telescopes on Earth. Figure 7 displays a mosaic of images, obtained by the SMART-1 scientific camera (AMIE), showing the SMART-1 impact site on the moon. Because of

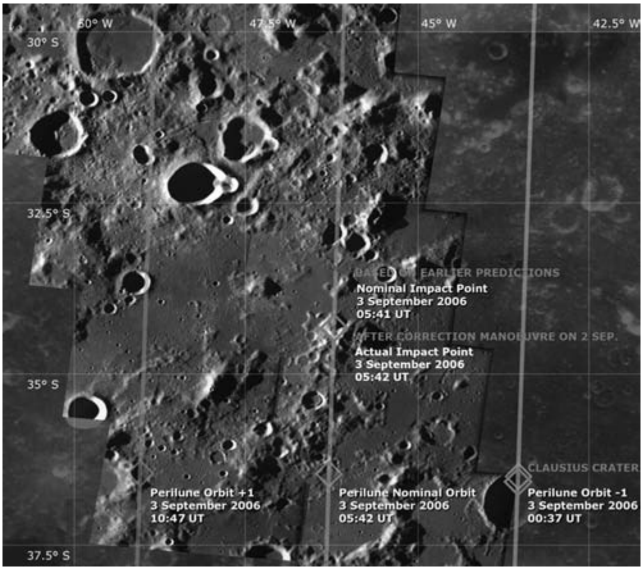


Fig. 7 Mosaic of images showing the SMART-1 landing site on the Moon. Credits: ESA/Space-X (Space Exploration Institute).

the correction maneuvers the nominal impact point was no longer accurate, and the spacecraft impacted, as predicted onto the ascending slope of a mountain of height about 1.5 km above the Lake of Excellence plain. Both the original (from before the correction maneuvers) and actual impact points are indicated in Fig. 7, together with the Clausius crater that triggered the correction maneuvers.

From the various observations and models, scientists will try to reconstruct what happened to the spacecraft and to the moon in the coming months. Predicted effects of the impact are the following:

1) Quick thermal flash and possible fireball due to remaining hydrazine onboard.

2) Yet another crater on the moon, perhaps some 5–10 m across, created by the high-speed slam dunk.

3) Dust and other material ejected off the moon, traveling some distance across the surface. Depending on the inclination of the mountainside that the spacecraft impacted on, the spacecraft could have ricocheted across the lunar surface.

D. Ground Observations

Professional and amateur ground observers all around the world have been watching the impact location to spot the faint impact flash and to obtain information about the impact dynamics and about the lunar surface excavated by the spacecraft. With the impact occurring nominally on 3 September 2006 at 05:42 UTC, observers from North and South America and the East Pacific were able to see the impact and were able to listen to its radio signal during night time.

Several radio telescopes measured the loss of signal, and recorded times remarkably in agreement with the last flight dynamics predictions. Through infrared observations with its newly installed infrared mosaic camera WIRCam, the Canada-France-Hawaii Telescope (CFHT) offered a stunning image of the SMART-1 impact, a very bright flash on the low-contrast landscape lit by earthshine. Figure 8 shows a mosaic of infrared images taken by CFHT. The figure shows the flash and the dust cloud that followed the SMART-1 impact. From a detailed analysis of the CFHT infrared movie of the variations after the flash, a cloud of ejected material or debris traveling some 80 km in about 130 s has been detected. It seems also that some ejecta or debris made it across the mountain. A careful analysis of the time evolution of the dust cloud, together with precise knowledge of the spacecraft dynamics at the time of the crash, should help to better understand the formation of ejecta following a lunar impact.

VI. Conclusion

The spacecraft orbit around the moon has been optimized for science successfully, both during the nominal mission phase and the extended mission phase. Coverage of a large part of the lunar surface—especially the South Pole regions—was achieved, at low altitudes and with good illumination, including opportunities for multiple observation of the same surface area.

Closer to the moon it became more difficult not to violate platform and payload constraints. The Mission Control Team commanded a solar panel offpointing in May 2006 to lower the solar panel temperatures. At low altitudes there was a concern for long blinding periods of both startracker cameras (at the same time),

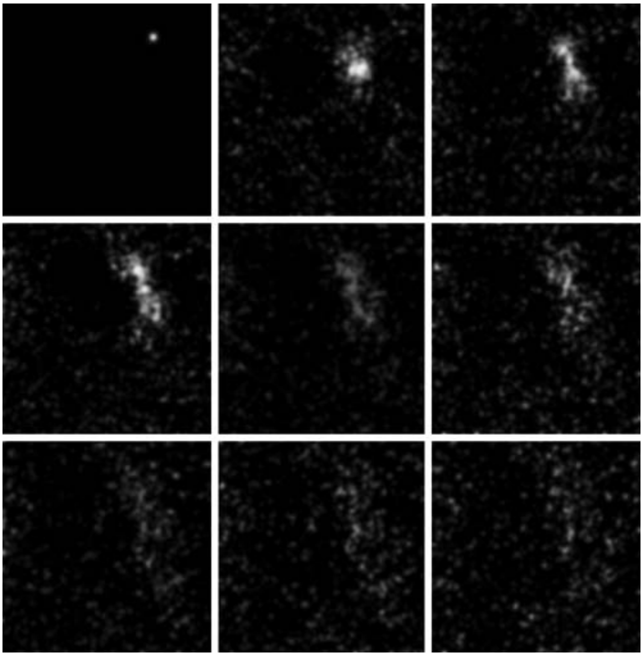


Fig. 8 Mosaic of images showing the flash and the dust cloud that followed the SMART-1 impact. Credits: Canada–France–Hawaii Telescope/2006.

and a special spacecraft pointing had to be implemented to clear at least one of the two startracker cameras.

There have been two main reboost phases during the science mission around the moon. In the first one, the electric propulsion engine was used to prepare the spacecraft orbit for the extended science mission, using all of the remaining propellant for the electric propulsion engine. In the second reboost phase, the hydrazine subsystem was used in an innovative way to do a series of small ΔV maneuvers. Without the second reboost phase, the spacecraft would have impacted the moon on the far side, away from ground contact and visibility.

Only 30 h before moon impact, a correction maneuver was executed (using the hydrazine thrusters), to maximize the probability of impacting as planned on the morning of 3 September at 05:42 UTC. Without this correction maneuver, the probability of impacting the rim of the Clausius crater, one orbit before the nominal impact, was high.

On 3 September, early in the morning at 05:42 UTC, SMART-1 ended its journey in the Lake of Excellence at 34.4ºS latitude and 46.2ºW longitude. Loss of signal was confirmed by the ESA ground station and several radio telescopes around the world. The impact was observed in infrared by the CFHT, offering a stunning image of the SMART-1 impact.

The SMART-1 legacy is a wealth of data, to be analyzed in the months and years to come. SMART-1 has mapped large and small impact craters, studied the

volcanic and tectonic processes that shaped the moon, unveiled the mysterious poles, and investigated sites for future exploration—a precious contribution to lunar science, at a time when the exploration of the moon is once again getting the world's interest.

Acknowledgments

The authors would like to acknowledge the excellent work that was done by all parties involved in the SMART-1 project, in particular the Swedish Space Corporation, the Société Nationale d'Etvdes et Construction de Moteurs d'Aviation (SNECMA), the Technical University of Denmark, the Project and Science Teams at ESTEC, the Mission Operations Team at ESOC, and the ground stations staff. The warm interest and support of Nottingham University, USGS, the observatories, and the public have been great. The success of reaching the moon, returning exciting lunar science, and of course the memorable SMART-1 impact on the moon has been thanks to the efforts of all.

References

[1] Milligan, D., Gestal, D., Camino, O., Bearman, P., and Koppel, C., "SMART-1 Electric Propulsion Operational Experience," *29th International Electric Propulsion Conference*, Princeton, NJ, Oct.–Nov. 2005.

[2] Khan, M., "SMART-1: Lunar Surface Coverage Analysis for the Actual Mission," ESA/ESOC, TN 44, Darmstadt, Germany, May 2004.

[3] Schwaller, D., "Memorandum: SMART-1 Thermal Predictions Prior to Crash," ESA/ESTEC, TEC-MCT/2006/DS, Noordwijk, The Netherlands, May 2006.

[4] Camino, O., Alonso, M., Gestal, D., De Bruin, J., Rathsman, P., Kugelberg, J., Bodin, P., Ricken, S., Blake, R., Pardo Voss, P., and Stagnaro, L., "SMART-1 Operations Experience and Lessons Learnt," *57th International Astronautical Congress*, Valencia, Spain, Oct. 2006.

[5] Bodin, P., "SMART-1 Operational Pointing Constraints," Swedish Space Corporation, S1-SSC-TN-2151, Solna, Sweden, June 2004.

Chapter 29

Phoenix Mars Scout Ultra-High Frequency Relay-Only Operations

Christopher A. Lewicki,* Joel A. Krajewski,† and Peter A. Ilott‡
Jet Propulsion Laboratory, California Institute of Technology Pasadena, California
and
Jason A. Dates§
Lockheed Martin Space Systems Company, Denver, Colorado

I. Introduction

RELAY missions are not a new idea. The Galileo probe, the Cassini–Huygens probe, and the Deep Impact "Impactor" relied on relay of their data via their "mother spacecraft." Even the space shuttles and the Hubble Space Telescope make use of the Tracking and Data Relay Satellite System to relay many aspects of their mission data. Relay communications have been used on previous missions to Mars as well. The Viking landers relayed much of their science data via 16 Kbps ultra high frequency (UHF) links and even entry descent and landing (EDL) 2 Kbps data to the Viking orbiters overhead [1]. The Sojourner rover received its commands and relayed data back to the Mars Pathfinder lander via a short-range relay link. The ESA's Beagle lander was to have been entirely controlled through the Mars Odyssey (initially) and the Mars Express (MEX) orbiters. The Mars Exploration Rover (MER) missions have demonstrated just how valuable a high-bandwidth return link via relay can be as they continue to explore Gusev and Meridiani [2]. Even for the Phoenix Mars Scout lander [3, 4], the concept of a relay is not new. In the Phoenix lander's original incarnation as the Mars Surveyor Program 2001 (MSP-01) lander, it was to have an all-relay landed mission using

*Flight Systems Engineer, Phoenix Project. Member AIAA.
†Surface Phase Lead, Phoenix Project.
‡Principle Telecommunications Engineer.
§Fault Protection Engineer.

the Mars Climate Orbiter (MCO) and its twin spacecraft, Mars Odyssey 2001 Orbiter (Odyssey) to achieve its command and control; additional data return support would have been provided by Mars Global Surveyor (MGS).

The subjects of the orbiter relay capability [5–8], the hardware and protocols employed [9–11], and the teamwork and operational processes required to support these multimission efforts [12] have been covered in detail elsewhere. This chapter will focus on the development, design, and planning necessary for a Mars surface mission to operate in an all-relay operational mode, without the safety net of a direct-from-Earth (DFE) emergency command capability, nor a direct-to-Earth (DTE) fault communications path.

II. Phoenix Landed Telecommunications Architecture

Prior to the MER landings, the MCO and Mars Polar Lander (MPL) failures motivated [13, 14] the addition of DTE capability to MSP-01 (Phoenix) in large part to provide data during EDL for forensic use in case of anomalies. EDL critical-event communications was a feature lacking from both MPL (and later Beagle), preventing confirmation of their suspected failure modes. Leading up to the Phoenix preliminary design review, several factors conspired for the return to the original MSP-01 configuration of an all-UHF, relay-only landed communications architecture. Many DTE/DFE design aspects related to the MSP-01 program were never fully implemented before its cancellation. In addition, the Earth-Mars range for the Phoenix mission was greater, further stressing the communication link performance. Once on the Martian surface, the data return requirements could not be demonstrated to be met with the X-band system alone, thus creating reliance on single-string hardware in an otherwise block-redundant and single-fault tolerant architecture. With the demonstrated performance of the UHF relay link between the MERs and Odyssey, the addition of the Mars Reconnaissance Orbiter (MRO) to the orbiting relays and a viable UHF-EDL communications strategy, a move back to the all-relay approach could be justified. Additional benefits included a significant mass savings on a lander beginning to be pushed to its limits, as well as a significant reduction in spacecraft implementation complexity. The resulting Phoenix telecom system (Fig. 1) consists of one antenna for EDL and two antennas for surface communications cross-strapped to block-redundant transponders and spacecraft electronics.

III. Relay Orbiter Support

The entirety of the Phoenix mission surface communication requirements will be satisfied through the use of either MRO or Odyssey as relay assets, although both will be used as available. Memoranda of Agreement are in place with both programs to ensure resources are allocated and planning is conducted to support the Phoenix mission [12]. The orbits of Odyssey and MRO allow for six to seven relay opportunities or overflights* per sol† to the Phoenix lander. Odyssey will

*An overflight is the term used to describe a single communications opportunity between the orbiting relay and the surface asset. The term *pass* is intentionally avoided to prevent communication with the DSN passes used by the orbiter to ultimately send the data back to Earth.

†A sol is one Martian day, or approximately 24 h and 39 min on Earth.

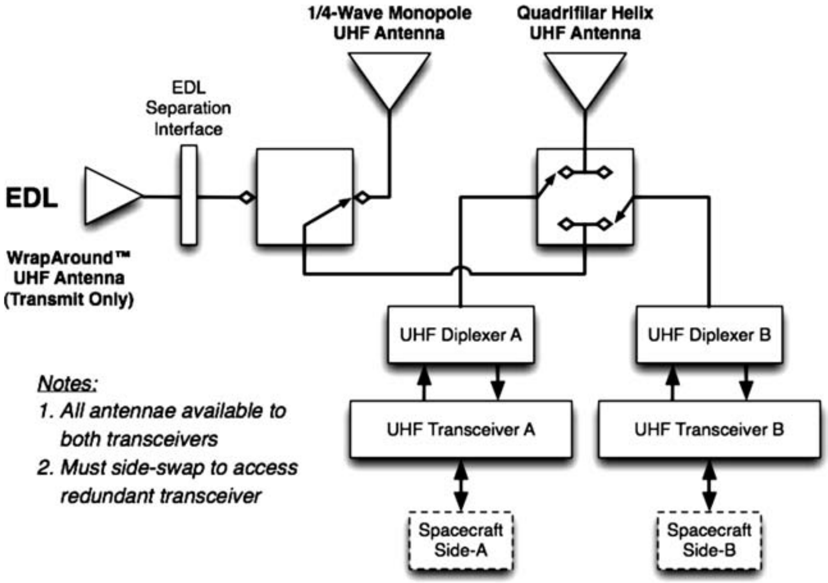


Fig. 1 Phoenix landed telecom block diagram.

attempt to establish a link during each overflight above a negotiated minimum elevation. MRO, on the other hand, will in general be limited to two attempts per sol since relay support diverts resources away from the MRO prime science mission. Routine Phoenix operations require two relay opportunities per sol among these eight to nine possibilities, with augmentation to three or four for improved science data return. While Phoenix is compatible with, and MER has made use of, the MGS for return of data, Phoenix is not pursuing this resource because of its inability to forward-link spacecraft commands. MER has also demonstrated compatibility with MEX for both return data and forward commanding. Phoenix is considering this as a contingency-only option that would be pursued only in the event a significant issue arose on either of its prime orbiters, compromising the redundancy in communications links.

Because Phoenix is a stationary lander, the relative orbiter geometry, relay opportunity timing, and link performance can be predicted for the rest of the mission after the final touchdown orientation is known. Assumptions about post-landed azimuthal orientation are eased by a Phoenix requirement to control its touchdown azimuth to within a few degrees. Dynamics of landing may induce a pirouette effect, perturbing the azimuth up to an estimated ±15 deg. The remainder of the final lander orientation is constrained by landing site selection and expected landing system performance. The exact timing of overflights is set no later than about eight weeks ahead of Phoenix EDL. To provide optimized communications coverage for the critical EDL events, both Odyssey and MRO will adjust their in-plane mean anomaly such that they are both in the vicinity of the Phoenix landing site at the time of EDL. As such, the time of the overflights for the rest of the mission will be set by this initial epoch. Because of the differing orbital periods as

well as the separated ascending nodes, it is infrequent that both orbiters will be overhead simultaneously. Outside of the initial dual-coverage of EDL, they will occasionally both be visible to the lander, and the frequency of this increases the more northerly the potential landing site. To ensure appropriate use of each of the orbiting assets, a strategic process [7] will be used to allocate and sequence support of Phoenix communications opportunities on each of the orbiters.

The analysis of the full mission communications coverage as just described has been performed for the purposes of performance estimation and verification of meeting mission requirements.

Direct support and coordination with the Deep Space Network (DSN) [15] is not necessary for Phoenix. While Phoenix is certainly in need of this service and is sensitive to its performance, this infrastructure overhead is indirectly coordinated on behalf of Phoenix by the orbiter programs.

IV. Overflight Geometry and Background

Potential landing sites for Phoenix have been down-selected to a region in the northern polar region of Mars, centered on 67.5°N latitude. Because of the sun-synchronous orbits of the relay, the distribution of overflights is effectively the same regardless of the longitude of the landing site. Plots of pass duration for various horizon cutoff masks, as well as orbiter elevations, as a function of a given local mean solar time (LMST) are shown in Figs. 2–5. Because of its northern polar location, Phoenix will enjoy increased access to the orbiting relays over what MER has experienced. The Phoenix lander will have direct line-of-sight geometry during every orbit for Odyssey and all but one orbit per sol for MRO. Both orbiter periods are approximately 2 h, but many orbits will be low on the horizon and at a long distance compared to overhead orbits, and therefore some of these do not have adequate link margin for successful relay.

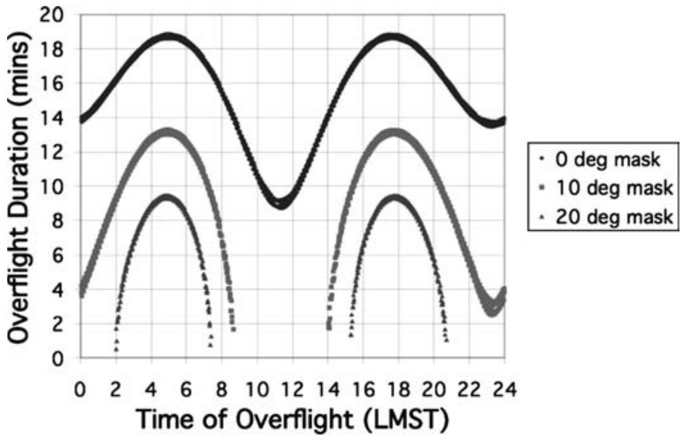


Fig. 2 Odyssey overflight duration vs local solar time, 67.5°N latitude. (See also the color figure section starting on p. 645.)

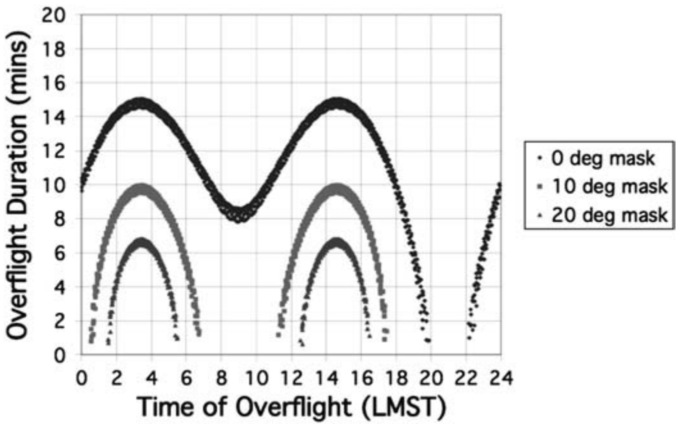


Fig. 3 MRO overflight duration vs local solar time 67.5°N latitude. (See also the color figure section starting on p. 645.)

V. Relay-Only Impacts on Fault Protection Strategy

There are many aspects of DFE/DTE-based spacecraft interaction that the operations community has become accustomed to as a matter of typical practice. When examining a fault-tolerant communications approach for an all-relay mission, a number of notable adjustments are necessary.

A. Relay Anomalies

When Phoenix spacecraft communication depends on an intermediary, the intermediary's problems become the Phoenix operations team's problems. These range from anomalies on the relay spacecraft, to support by the DSN, and even

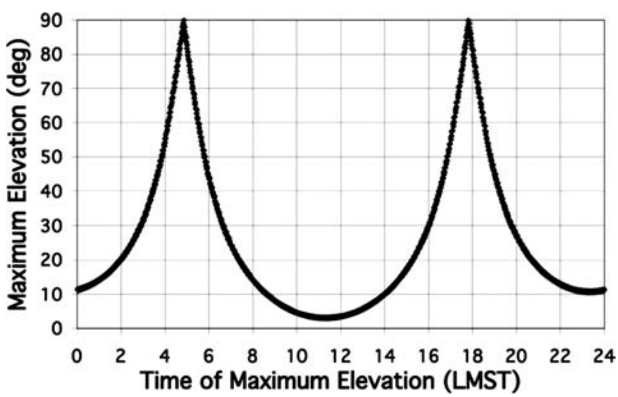


Fig. 4 Odyssey maximum elevation vs local solar time, 67.5°N latitude.

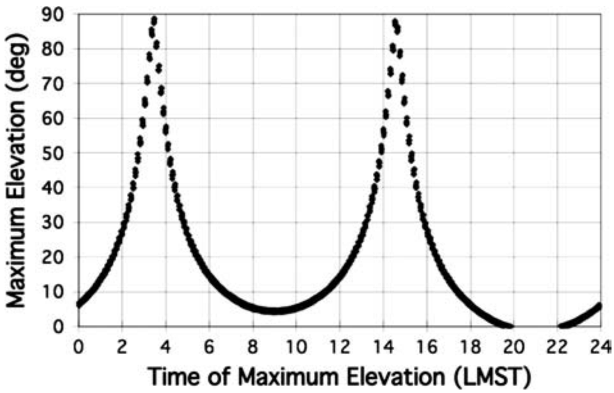


Fig. 5 MRO maximum elevation vs local solar time 67.5°N latitude.

include personnel issues on the relays' operations teams. The ideal communications and fault protection approach should be as insensitive as possible to any of these problems. A relay anomaly should not automatically induce an anomaly on the lander. Phoenix will employ two operational mitigations: 1) configure onboard fault protection to declare a fault due to lack of successful relay only after several sols have elapsed (uplink loss), and 2) maintain "run-out" sequences on the lander to be executed during these few sols of no ground contact.

In addition to the mission requiring the lander to be self-supporting for a short duration without ground interaction, the operations team will also plan for diversity in relay orbiter coverage, not placing all of the relay responsibility on a given orbiter. Should a relay anomaly prevent its support of lander communications, the next alternate orbiter pass would allow for short-term or long-term adjustment of the lander communications strategy if needed.

B. Spacecraft Fault Communications Rate

1. Control of Communications State

A common practice for configuration of spacecraft during fault protection responses is to place the communications system in a known, safe, and robust uplink and downlink state. Usually this will involve using a communications configuration with the highest link margins, and therefore lowest data rates, to provide the best chance of successful receipt of commands and return of data. Further, the most robust configurations typically utilize antennas with lower gain and wider beam-widths.

For Phoenix, the orbiter will "hail" the lander using the Proximity-1 protocol [9, 11], and establish the configuration of the link without knowledge of the state of the lander. In this implementation of the protocol, the orbiter controls the data rate at which the lander transmits. This may result in lander anomalies being discovered in a nominal communications mode, with link rates typical for daily routine operations (128 Kbps return rate, 8 Kbps command rate). If the high-data-rate

link can be supported in spite of the fault, the Phoenix mission planners may have a wealth of data to aid in diagnosis. This is an interesting paradigm shift, as it may often mean that the data management process necessary to keep the science data returning to ground operators will not necessarily be interrupted by minor spacecraft anomalies.

While high-data-rate fault communications may often be achievable, there are situations were the operations team may prefer a lower communications rate, likely for one of two reasons: 1) the anomaly the spacecraft is experiencing is related to the communications subsystem itself, and the lower data rates may be necessary to achieve a robust link, or 2) the lower data rates will allow for more frequent opportunities to interact with the spacecraft, taking advantage of the lower-elevation, lower-link-margin overflights.

To achieve this change in relay configuration, *it is the orbiters* that must be commanded to change their state, not the lander. This configuration change will have to be accomplished after the ground detects a fault (by received lander data) or suspects a fault may have occurred (by lack of communications with the lander), and must be achieved within the turnaround times that are possible with the latencies in operations decision processes, DSN passes with the orbiters, and upcoming relay opportunities with the lander. Alternative relay configurations are also discussed in Sec. V.E.

2. Command Screening

An advantage of a spacecraft fault response that places the communications subsystem in a distinct fault communications state is that the command uplink rate can be used to screen commands that were intended for a non-faulted spacecraft. In this configuration commands transmitted to a spacecraft at a higher nominal rate will not be received by a spacecraft that is listening at a lower rate. The inverse is also true, and has its own advantages.

With the implementation of a Proximity-1 link controlled by the orbiter, there is no mechanism to achieve this "command screening" by data rate alone. Because of this, Phoenix has modified its flight software to indicate to the command and sequence engines when it is in a safe mode, and the operations team must always check for this state when attempting to execute sequences that should not be executed if such an anomaly has occurred.

The Phoenix implementation of this feature has its own advantages in that a single command load can be responsive to multiple-spacecraft states, without the need to reconfigure the command process and radiate separate command data for the possible spacecraft states. As an example, a safe mode condition could be commanded to conserve energy, whereas a nominal condition could continue as previously planned.

C. Real-Time Spacecraft Interaction

Interplanetary spacecraft operations must always accommodate round-trip light-time delays when interacting with a spacecraft. For complex or critical activities, a common operations approach is to have "GO/NO-GO" opportunities to confirm a spacecraft or instrument state and then, within minutes, send a command in

response to that state. While a relay link does not necessarily prevent this type of interaction, in the case of Phoenix and its support by MRO and Odyssey, it prevents it from being practical. With orbiter overflight durations limited to no more than 16 min at most, and round-trip light-times to Mars being a minimum of 30 min during the Phoenix mission, a given overflight would be finished before the operations team had an opportunity to command a response. This is without considering the overhead on preparing and radiating a command load to the orbiter, although special measures could be implemented to require only a single real-time command to the orbiter to relay a pre-loaded onboard response. Utilization of this technique for Phoenix is limited to successive overflights, and therefore the best possible turnaround time is measured in hours, not minutes. While this approach would provide the capability for GO/NO-GO cycles, it does so much less efficiently (in turnaround time) than a DFE/DTE link would provide for.

D. Commanding "In the Blind"

The operations team can never be certain of the state of the spacecraft prior to sending commands, and hence commanding is always "in the blind." Measures such as the safe mode command screening described previously are necessary to mitigate the potential negative impacts of this "feature." Because of this, two things become of paramount importance, as noted in the following.

1. Unintended Consequence of Commands

Given that the spacecraft could be in a different configuration than expected, constant awareness by the operations team is necessary to understand the potential effect of the commands they send. This not only includes the safe mode state previously mentioned, but also possible faulted states for payloads and other hardware. For Phoenix, similar command screening is implemented for the payloads, effectively creating a safe mode state for each, to prevent unintentional commanding from uplinked sequences. In addition to this onboard screening capability, a ground-tool screening process is implemented through verification of flight rules that have been identified by the design and operations teams. Automatic screening can verify the majority of these flight rules, although a complex subset must be verified manually.

2. Deterministic Behavior

Keeping autonomous fault response timelines as deterministic as possible will aid in the spacecraft operations teams' ability to diagnose and respond to potential faults. Any state that may be asserted by onboard fault protection must be completely unambiguous and easily reproducible on the ground. While this is true for any spacecraft, it is even more important for a relay-only surface mission, where knowledge of the spacecraft state (i.e., awake and listening vs asleep and not listening) is critical to properly plan relay passes. For this reason, a ground-in-the-loop downlink loss response (relying on continued ground notifications to execute a pre-planned sequence of telecom configurations) was abandoned for Phoenix, in favor of an uplink loss response that executed all possible configurations.

This single uplink loss behavior executes not only recovery strategies related to loss of uplink, but those related to loss of downlink as well.

E. Communications Failures

1. Scenarios

The inability to make use of the Cincinnati Electronics 505 (CE-505) transceiver implementation of the Proximity-1 protocol in the event of an anomaly has consequences that will now be discussed. In a two-way reliable communications session, both ends of the link must work, or no information is exchanged. The inability of the surface radio to receive transmissions from the orbiter means that the lander will not receive the acknowledgment (ACK) confirmation that its transmissions are being received. Likewise, if the lander is unable to successfully transmit data, the orbiter will not be able to send command data to the lander, as it could never confirm receipt.

Failure of either direction of the link at the UHF hardware level can manifest itself in two scenarios. A failure during an established link would result in both ends of the link repeatedly transmitting two Proximity-1 frames of data, never successfully getting ACK of successful receipt on the other side. A persistent failure (perhaps staying failed after the aforementioned failure mode) would prevent a new link from ever being established. The orbiter would attempt to hail the lander and establish the communication session, and the lander would either not be able to hear the orbiter, or it would be unable to respond.

2. Autonomous Responses to Failures

If a failure of the telecommunications system is directly observable by the spacecraft (e.g., the UHF transceiver fails to power up), the immediate response will be to use the redundant transceiver hardware. In the case of Phoenix this means a series of resets and re-attempts that, if the failure persists, will quickly change the spacecraft over to its redundant side of hardware.

Less obvious failures may occur, and where they are not directly detectable by the spacecraft, it must eventually realize that "something is not working" and take similar measures to access the redundant hardware. The fault response implemented for this purpose is the often-used uplink loss response. In this fault case the spacecraft has a configurable countdown timer that is reset by direct command. If the spacecraft operators fail to issue this command before the timer expires (either through unintended omission or the inability to command), the specialized uplink response is engaged until command capability has been re-established and the response is disabled.

In the unfortunate event that the failure mode is in the reliable link itself (either due to the lander or the orbiter), additional fault tolerance can be added by augmenting an ultimate safety net mode in which the spacecraft will eventually attempt one-way, "unreliable" or open-loop links. The current Phoenix concept incorporates a cyclic pattern of receive-only, one-way transmit, and reliable modes, through all available antennas. The final definition of this mode will have to be balanced against spacecraft resources, such as power availability and thermal environment.

3. *Ground Responses to Failures*

With limited opportunities for contacts between the orbiter and lander, any communications opportunities necessary for fault response must be planned for *in advance*. Both the orbiter schedule, as well as the onboard communications schedule, must account for fault communications opportunities. Sequencing additional orbiter overflights will have no effect if the lander has not previously been configured to make use of them.

The first line of defense in Phoenix fault response design is to enact a communications schedule stored in an onboard table, periodically updated by the operations team. This table may include use of the "nominal" communications opportunities, or it may be adjusted to use more or less frequent overflight opportunities, depending on available spacecraft resources and preferences of the operations team. This strategy will be employed for most faults and will provide for minimal delay in recognizing and responding to a spacecraft fault. A similar, albeit more complex, approach was successfully implemented on MER.

4. *Dead-End Behavior*

The failure to correctly synchronize the timing between the lander and orbiter communications periods is perhaps the most challenging situation to robustly design for, and typically represents "the end of the road" in mechanisms the spacecraft will employ in its efforts to re-establish communications. This disconnect between Phoenix communications attempts and orbiter availability may arise for several reasons: 1) a corrupted, improperly prepared or completely executed safe mode communications table, 2) substantial drift or loss of onboard absolute spacecraft time, or 3) substantial deviation in the orbital parameters of the orbiters. In response to these potential faults, Phoenix has chosen to implement a fixed time-step mode in which a configurable, fixed sleep duration and wake (listen) duration can be applied between successive communications attempts. The intent of this approach is to achieve communications attempts at different local solar times on every attempt, eventually walking around the clock and landing on a time when an orbiter will be overhead. Achieving this in a timely manner is challenging due to constraints on available energy and the level of conservatism used when determining the ability to communicate with an orbiter. The Lander must "sleep" most of the day to conserve power and recharge batteries, which limits the time during which it can listen for a signal from an orbiter. For conservative worst-case assumptions with respect to performance of the telecommunications system, analysis has shown that contact with the spacecraft can be achieved within one sol for a majority of the cases, but a worst-case misalignment may cause recontact to take upwards of one week. These cases represent a small fraction (less than a few percent) of all cases, and more nominal performance parameters reduce the severity of these outliers.

An alternative to the fixed time-step approach is to implement a mode in which the UHF transceiver is powered on in receive-only mode when the spacecraft is in a faulted state. The receive-only mode of the transceiver uses only 10% of the power that the transmitting mode requires, and the next time an orbiter is available, a communications session could be established. With the high heritage that Phoenix has from the MSP-01 spacecraft design, implementation of this approach

requires nontrivial architecture changes and the project has decided that the implementation risk of this approach is greater than the risk of potential recovery delays introduced by the fixed time-step mode. The 2009 Mars Science Laboratory Rover is being designed from the outset to include this capability as well as some useful extensions to it.

VI. Phoenix UHF Communications Test Program

The Phoenix program has implemented an extensive verification and validation program to ensure the successful operation of the lander in its use of the relays. This program began with testing of an engineering Phoenix Lander Simulator incorporating a CE-505 transceiver against the MRO flight spacecraft and its Electra UHF transceivers in the various modes that Phoenix plans to use in-flight. Thorough simulation and subsequent measurement of the approximate UHF antenna patterns attainable by the helix and monopole antennas were conducted at the Space and Naval Warfare antenna test range in San Diego, California. These measured antenna patterns were incorporated, with margin, into simulations of the expected in-flight performance of the telecommunication subsystem to derive predictions for performance during the mission. In the assembly test and launch operations process of building up the Phoenix lander, the flight transceivers have been tested at several opportunities with not only orbiter simulators, but also the actual spacecraft test beds of Odyssey and MRO, allowing for a “test like you fly” configuration that was not achieved for MER. Operational readiness tests with the spacecraft operations team and flight-like command and data configurations will prepare the operators for the nuances of relay operations described herein. Lastly, in-flight tests conducted by MER, Odyssey, and MRO on behalf of the Phoenix project will give further confidence in the performance of the system.

VII. Challenges in Routine Operations

Forward and return link latency is governed by a number of variables ranging from human and computer processing time to margin time for event boundaries, to the laws of physics and the speed of light. Part of the strategic communications processes [5, 12] will be to determine these latencies for all potential overflights. The operations team must complete all of their present sol analysis and performance assessment activities, plan for the next sol, and uplink it to the relaying orbiter within these constraints. Whereas MER started with a 16-h process, enabled in part by its DFE command capability and reduced latency, Phoenix will have to work within tighter constraints, starting at 14 h, and trending downward as more activities are added to the day.

VIII. Special Considerations

Time synchronization of a spacecraft through a relay involves applications of technologies not yet demonstrated in flight [16]. In addition, these new protocols are not supported by the heritage CE-505 UHF transceiver onboard the Phoenix lander. Phoenix has examined the performance of its hardware and the initial

assessment is that the expected clock-drift during the 90-sol prime mission (expected to be 40 s or less) is well within the timing requirements required to operate the mission. While certain fault scenarios may challenge this assertion, a number of techniques are available for both coarse (10s of seconds) and fine (seconds or less) time correlation. These include observation of communication timing differences between the well-time-correlated orbiters and the lander, to sequenced corrections based on observations of either the sun or mars' moons, Phobos and Deimos.

IX. Conclusion

The Phoenix mission is well situated in the timeline of planetary exploration to take a significant step forward in routine surface operations via relay of commands and data through orbiting spacecraft. The experience gained from prior missions including MER and Odyssey, the resources available to Phoenix in the form of well-tested relay assets, and attention to the unique needs of a surface relay-operated mission will ensure success in operating the Phoenix mission.

Acknowledgments

This work was supported by the Phoenix Project of the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. Some of the work described in this publication was carried out at the Jet Propulsion Laboratory.

References

[1] Lee, B. G., and Porter, J. D., "Design and Implementation of Viking Mission," AIAA 13th Annual Meeting and Technical Display Incorporating the Forum on the Future of Air Transportation, Washington, DC, 10–13 Jan. 1977, p. 9; AIAA Paper 1977-270.

[2] Taylor, J., Makovsky, A., Barbieri, A., Tung, R., Estabrook, P., and Thomas, A. G., "Article 10: Mars Exploration Rover Telecommunications," JPL Technical Rept. Service, 06-1429, Jet Propulsion Lab., Pasadena, CA, 2005, http://hdl.handle.net/2014/39934.

[3] Shotwell, R. "Phoenix: The First Mars Scout Mission," *Acta Astronautica*, Vol. 57, July–Oct. 2005, pp. 121–134.

[4] Smith, P. H., "The Phoenix Mission to Mars," *IEEE Aerospace Conference,* Vol. 1, 6–13 March 2004, p. 342, doi: 10.1109/AERO.2004. 1367617.

[5] Edwards, C. D. Jr., Barbieri, A., Brower, E., Estabrook, P., Gibbs, R., Horttor, R., Ludwinski, J., Masel, R., McCarthy, C., Schmidt, R., Theisinger, P., Thorpe, T., and Waggoner, B., "A Martian Telecommunications Network: UHF Relay Support of the Mars Exploration Rovers by the Mars Global Surveyor, Mars Odyssey, and Mars Express Orbiters," *55th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, Vancouver, Canada, 4–8 Oct. 2004.

[6] Edwards, C. D. Jr., Arnold, B., DePaula, R., Kazz, G., Lee, C., and Noreen, G., "Relay Communications Strategies for Mars Exploration Through 2020," *56th International*

Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law, 2005.

[7] Lewis, J. A., "Mars Odyssey Relay Operations Development," *IEEE Aerospace Conference*, 5–12 March 2005, pp. 4200–4211, doi: 10.1109/ AERO.2005.1559724.

[8] Makovsky, A., Barbieri, A. J., and Tung, R., "Article 6: Odyssey Telecommunications," Jet Propulsion Lab., Pasadena, CA, Oct. 2002.

[9] Barbieri, A. J., Butman, S., Danos, M. J., Greenberg, E., Ilott, P. A., Kazz, G. J., Torgerson, J. L., Vaisnys, A., Adams, W. R., Johnson, C. E., Dapore, M., and Merz, D., "Development and Flight Performance of CCSDS Proximity-1 on Odyssey and the Mars Exploration Rovers," *IEEE Aerospace Conference*, 5–12 March 2005, pp. 1444–1454, doi: 10.1109/RERO.2005.1559435.

[10] Edwards, C. D. Jr., Jedrey, T. C., Schwartzbaum, E., Devereaux, A. S., DePaula, R., Dapore, M., and Fischer, T. W., "The Electra Proximity Link Payload for Mars Relay Telecommunications and Navigation," *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, Bremen, Germany, 2003.

[11] *Recommendation for Space Data System Standards, Proximity-1 Space Link Protocol—Data Link Layer, Recommended Standard*, Consultative Committee for Space Data Systems CCSDS 211.0-B-4, July 2006.

[12] Gladden, R., Hwang, P., Waggoner, B., McLaughlin, B., Fieseler, P., Thomas, R., Bigwood, M., and Herrera, P., "Mars Relay Coordination Lessons Learned," *IEEE Aerospace Conference*, 5–12 March 2005, pp. 177–190, doi: 10.1109/AERO.2005.1859311.

[13] Albee, A., Battel, S., Brace, R., Burdick, G., Burr, P., Casani, J., Dipprey, D., Lavell, J., Leising, C., MacPherson, D., Menard, W., Rose, R., Sackheim, R., Schallenmuller, A., and Whetsel, C., "Report on the Loss of the MPL and DS-2 Missions," Jet Propulsion Lab., California Inst. of Technology, Pasadena, CA, 2000.

[14] Whetsel, C., et al., "The Return to Flight Criteria for the Mars Surveyor 2001 mission," Jet Propulsion Lab., California Inst. of Technology, Pasadena, CA, 2000.

[15] Hill, R., Jr, Chien, S., Smyth, C., and Fayyad, F., "Planning for Deep Space Network Operations," *Proceedings of the AAAI Spring Symposium on Integrated Planning Applications*, Palo Alto, CA, 1995.

[16] Antsos, D., "Mars Technology Program Communications and Tracking Technologies for Mars Exploration," *IEEE Aerospace Conference*, 5–12 March 2005, doi: 10.1109/AERO.2006.1655778.

This page intentionally left blank

Chapter 30

From Mission Concept to Mars Orbit: Exploiting Operations Concept Flexibilities on Mars Express

Jonathan R. Schulster*
Scisys, GmbH, Darmstadt, Germany
Michel Denis† and Alan Moorhouse‡
European Space Agency, Darmstadt, Germany
and
Erhard Rabenau§
Nova Space Associates, Ltd., Bath, England, United Kingdom

I. Introduction

THIS chapter presents the development of the Mars Express (MEX) operations concept within a historical framework. Starting from the loss of the Mars 96 Russian orbiter and the birth of Mars Express, the chapter leads on to the sizing of the design-case mission [at Astrium Space SA (ASTRIUM) in 1999], and through to the European Space Operations Centre (ESOC) Operations Concept Workshop in February 2000. It goes on to describe the evolution and changes evident at the Mission Planning and Operations Concept workshop in April 2002 and finally experience from the flight domain covering 2003–2006. The operational costs and their evolution are not covered.

The next section will present a brief history of the Mars Express mission, its origins in the failure of Mars96, and the implications of these origins in terms of operational orbit, payload accommodation, and the development life cycle.

Section III will describe the operations concept as it was initially designed in 1999/2000 during the specification and development of the spacecraft and early

*Flight Control Team/Mission Planner, ESA Operations—Flight Operations Department.
†Spacecraft Operations Manager, Mars Express, European Space Operations Centre.
‡Deputy Spacecraft Operations Manager, Mars Express, European Space Operations Centre.
§Chief Mission Planner.

preparation of the ground segment. At this stage, the concept was intended to converge toward fully detailed mission definition before launch.

Section IV will describe the mission operations and planning concept as defined by 2002, during the final integration of the ground segment. It will highlight the evolution made in some areas of the mission definition and operations concept since Phase A/B (2000), and identify those aspects that still remain to be solved in flight.

Section V describes the evolution in the operations concept over the life cycle of the Mars Express mission, including the changes in mission design and resources with significant impact on the mission design. In order of impact, this will cover the restrictive power capability onboard the spacecraft, the move to round-the-clock ground station coverage, and modifications to the science mission requirements.

The stretching of flexibilities in the operations (and mission planning) concept to cope with the changes in the demands of the mission are covered in Sec. IV.B. The flexibilities provided by the planning concept could be exploited in various ways to maximize the mission return within the constraints and resources provided.

The conclusions highlight the most important lessons learned from the Mars Express mission in terms of the development of its operations concept and how this might be useful to the definition of other missions in the future.

II. Birth of Mars Express

Following the success of the Phobos mission in 1989, the Russian Space Agency focused planetary exploration on the planet Mars, driven primarily by an interest in terrestrial-planet comparative climatology, and its eventual target as the first planet for manned space missions. Mars96, weighing 6700 kg and with a total payload of 550 kg, was launched on 16 November 1996, but because of a failure in the Block-D upper stage reentered the atmosphere and fell into the Pacific. Among a total of 29 payload instruments were included Analyzer of Space Plasmas and Energetic Atoms (ASPERA), Observatoire pour la Minéralogie, l’Eau, la Glace et l’Activité (OMEGA), Planetary Fourier Spectrometer (PFS), Spectroscopy for Investigation of Characteristics of the Atmosphere of Mars (SPICAM) and High Resolution Stereo Camera (HRSC), which were all selected later for inclusion in the payload of Mars Express. An illustration of the Mars96 spacecraft is shown in Fig. 1.

The background and history of how the ESA conceived Mars Express is not specifically part of this chapter; however, it can be summarized as a view by ESA that Mars96 represented “European effort, worth re-flying” and that this desire could be met by the accommodation of some of the payload instruments from Mars96 on Mars Express.

The operations concept for Mars96 was radically different from that eventually selected for Mars Express. Some of these differences had major impacts on the design and operations of Mars Express.

Several of the instruments on Mars96 were mounted on a steerable, dedicated instrument platform. The HRSC (imaging camera) and OMEGA [infrared (IR) spectrometer] shared this platform. Conversely on Mars Express all instruments were fixed, body-mounted with field of view in the +*z*-axis direction (nadir face during science pointing)—with the exception of SPICAM (UV/IR spectrometer)

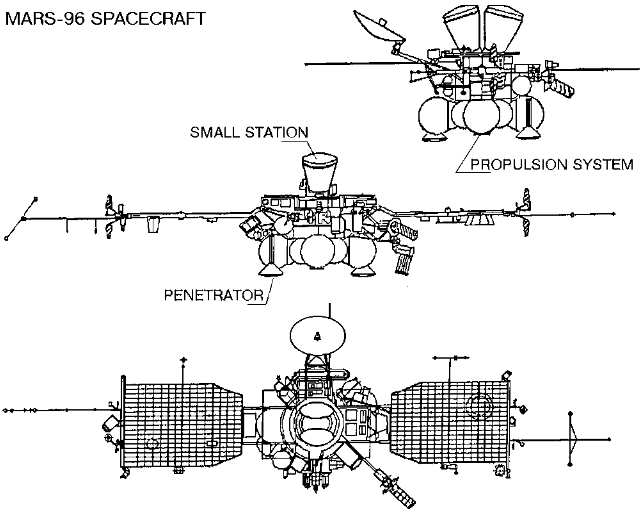


Fig. 1 Mars 96 spacecraft.

that additionally has a slit aperture for solar occultation observation in the $+y$ radiator face. Several instruments use scanners and mirrors to provide the necessary observational field of view; however, basic pointing is provided by re-orientation of the entire Mars Express spacecraft.

A single, body-mounted high-gain antenna (HGA) of 1.6-m diam is provided for communications with Earth ground stations. The operations concept was therefore restricted by the fact that orientation of the spacecraft could either provide a pericenter nadir pointing for science observation or fixed pointing of the HGA to Earth, but not both at the same time.

A further difference in the fundamental mission design of MEX vs Mars96 was the orbit. That of Mars96 would be a relatively low Mars orbit, roughly circular in dimension at a few hundred km altitude. That of Mars Express was limited by the available fuel after capturing into planetary orbit, and was chosen to be nominally 10,500 $\times$ 250 km pericenter. With a period of about 6–7 h, this would be asynchronous with the Earth day, leading to inevitable overlap on some orbits between the communications window with the single ground station (ESA New Norcia 35-m antenna in Australia) and observational pericenter passes at Mars.

Budgetary and schedule constraints aiming at a “flexi” mission and a launch in 2003 resulted in the selection of a “cut down” Rosetta Comet Mission (ROSETTA) platform as the bus for Mars Express. The effort for a European interplanetary platform, started several years before in view of a comet visit, had in the late 1990s reached a mature design leading to the final development phase for the flight and ground segments. Without the possibility of reusing the ROSETTA avionics, virtually unchanged, it is unlikely that Mars Express would have been possible.

Finally, a major although nontechnical contributor to the decision “Europe goes to Mars” was the impressive success and outreach of the NASA rover Sojourner in July 1997, also largely relayed by the Internet in its early blooming age.

III. Early Operations Concept (2000)

The following section outlines the status and understanding of the operations concept for the Mars Express mission as presented at the Operations Concept Workshop in February 2000, specified early during the development phase for the ground segment.

A. Mars Express Operations Scenarios

ASTRIUM presented the "Big Picture" of the mission design in 1998, at the time of the Preliminary Design Review. In Fig. 2 the implications of this mission design for the operations concept drivers, as understood by ESOC at the Operations Concept Review in February 2000, are overlaid on the original diagram. These mission drivers and their implications are described in detail in the next section.

These mission drivers were used to define operations scenarios that allowed development of the solutions and baselines to be defined that would comprise the operations concept for the mission.

Following a launch from Baikonur on a Soyuz launcher in May/June 2003, a cruise phase lasting six months would culminate in Mars orbit insertion around Christmas 2003. The initial capture orbit would be reduced to an operational orbit of about 10,550 × 250 km altitude, using either the bipropellant main engine or aerobraking in the Martian atmosphere.

In each operational orbit with a total period of 6–7 h, one pericenter nadir pointing pass would be used for scientific observation. About 8 h each day would be reserved for Earth communications, to upload commands for future execution and

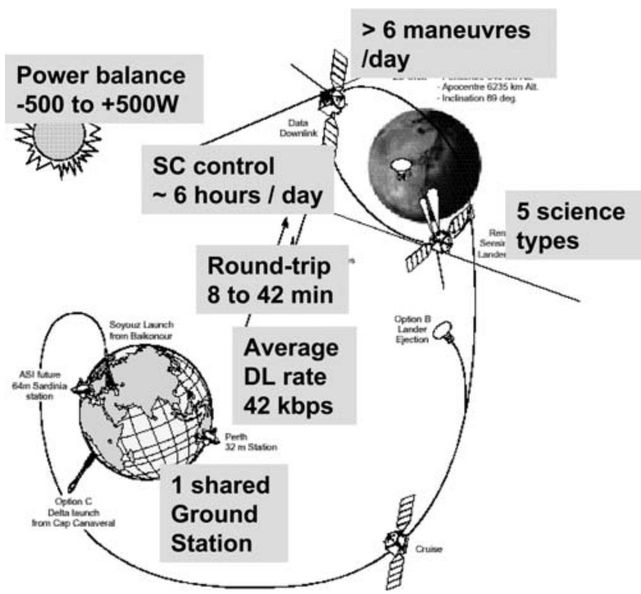


Fig. 2 "The Big Picture"—ESOC view of the operations concept (2000).

for download of all science and housekeeping data. The instantaneous power balance on the bus could vary between eclipse/nadir science observations and recharge cycles by as much as 1 kW. A typical daily cycle could include a wheel momentum off-loading phase, three nadir science passes, and a single 8-h communications Earth pointing pass.

B. Mission Operations Drivers

This section describes the main features, mission drivers, impacts, and operational solutions adopted for the Mars Express mission. The large interplanetary distances involved drives the signal round-trip time and impacts on the necessity for offline commanding via a mission timeline (MTL) with time-tags in telecommand (TC) files.

Solar conjunction periods without ground contact for up to 30 days (worst-case assumed in 1998/2000) created the need for onboard autonomy and the so-called flying control center comprising potentially a special MTL, TC files (macros), software files for safe mode reconfigurations.

The variable Earth–Mars distance results in variability in solar array power with a factor of 1.4 between Mars aphelion and perihelion. This generates an operational requirement to prevent battery negative drift accumulated over a series of orbits, in turn necessitating ground battery/energy monitoring, in particular during eclipse seasons, and a power modeling and prediction capability at mission planning level.

For the payload instruments, the multiple pointing requirements and complex duty cycles drive the spacecraft attitude profile. The requirements for contradictory system use between pointing types for nadir, off-track, stellar, and radio science would be resolved via a Master Science Plan (generated by the Project Science Team) and a complex Mission Planning System. The large volumes of data generated by the optical payloads impacted on the design of the links to the solid-state mass memory (SSMM), storage size available, and the need to preserve data online (on ground) for as long as 10 days.

The telecommunications relay and Beagle-2 visibility times required short-period closed-loop with Earth, particularly in the event of a Lander contingency. A strategy involving scheduled Lander access ultrahigh frequency (UHF) during nadir observations was adopted.

Sharing of primary ground facilities with other missions (ROSETTA) gave restricted ground contact times and was the main mission driver for priority handling between missions. Operationally this gave rise to the need for a “fair data return” policy between the instruments, and a “criticality wins” scheme for assigning pass time. The baseline of one ground station, with an 8-h maximum pass each day implied that no backup ground station would be available for routine operations. The mission must be tolerant to one missed station pass, and therefore a baseline of 48-h duration timeline onboard was taken as the design case.

Limited budgetary constraints implied a reduced-size operations team, with an impact on manning and effort. Operationally this meant minimizing real-time spacecraft operations, and any unplanned interactions with prime investigators (PIs) for science observations. A weekly planning cycle was adopted with five days per week working hours support for spacecraft engineers and mission planning.

C. Management of Operations

The management of operations on the Mars Express mission involved various actors, as described in Fig. 3. The initial management concept, as envisaged in 2000, was classical, involving five primary actors, and five direct day-to-day or weekly interaction routes between them.

The Flight Control Team (FCT) at ESOC would be responsible for mission planning, uplink of the telecommands to the MTL, and control of the flow of science/housekeeping data back to Earth from the spacecraft.

Requests for attitude changes and schedules for orbit maintenance would be sent to Flight Dynamics (FD), who would distribute maneuver definitions back. Ground Operations would interact with FD for tracking and antenna pointing, and with the FCT for ground scheduling and station and communications support.

The PIs and their science operation teams would send payload operations requests (PORs) to ESOC and receive data directly from a dedicated server at ESOC.

D. Deep Space Operations Requirements

1. *Deep Space Ground Protocol*

It was recognized by ROSETTA, and adopted by Mars Express, that for commanding Command Operations Procedure (COP-1) was not suitable, and there was a need for secure file transfer (FT) protocol. As a minimum, the uplink capacity would be twice the total of telecommands required until the next pass.

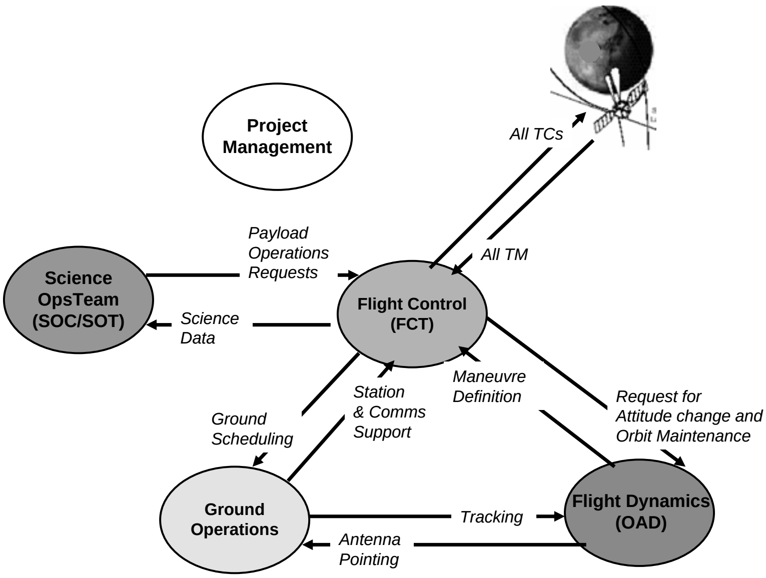


Fig. 3 Mars Express actors of operations (2000).

For telemetry, downlink priorities would be assigned per data type (or SSMM packet store), such that downlink order could be predictable and controllable. Telemetry (TM) files onboard would be erased periodically, coordinated with ground reception of file contents. Timing identification for all packets was required.

Telecommunications aspects included the pass strategies for routine communications (Fig. 4), programming of TM dump windows, decoupled from commanding uplink, and bit rate changes (for telemetry) to optimize the downlink. Science planning was to be based on the integration of the bit-rate over future passes. For safety critical situations, safe mode would always ensure some communications.

2. *Onboard Management*

The provision and maintenance of onboard software is seen as the main way to cope with mission complexity, duration, unforeseeable events and incidents. This implies that efforts are made to maximize flexibility of architecture to allow in-flight software solutions. The spacecraft manufacturer prepared a list of foreseen fault management routines that would be onboard to cope with a mission critical failure.

Programmed operations utilizing automation would include TC, TM, spacecraft monitoring, reaction to anomalies and onboard software maintenance (OBSM). The onboard schedule (MTL) provides the ability to run procedures without ground interaction.

3. *Optimizing Contact with the Spacecraft*

The pass strategy for Mars Express included two major issues to be decided: First, how daily contact (roughly 8 h) would be defined within the variable 9–13-h

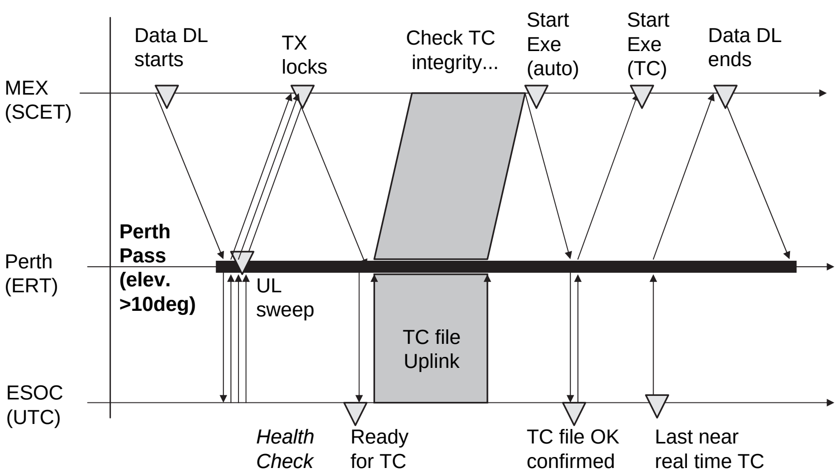


Fig. 4 Scheduling timelines: two worlds in parallel.

geometric visibility? Second, when the ground station pass conflicted with a pericenter passage at Mars, what priority would be assigned to science vs communications? The actual commanding window depends on Mars-Earth distance and logic of operations — how many round trips needed to finish near real-time procedure. A safety margin is required before leaving the spacecraft alone for at least 16 h. The issue of extending the telemetry dump beyond 8 h, if the spacecraft was still visible, was primarily one of cost; station costs were assigned on the basis of an 8-h shift, as well as the planned manning of the control center (spacecraft controller).

The timeline for operations would have to cope with "two worlds in parallel," with the continuous coordination of events onboard (Mars time) with those on ground (Earth time). This would eventually entail the creation of a complete ground station planning system, mainly driven by the events and activities at Mars, but in 2000 it was only foreseen to ever use a single ground station.

The parameters controlling the Mars Express mission include telemetry and telecommands volume, pointing directions per day, the time to acknowledge one TC, time taken to load/dump a file of 1000 TCs, spacecraft elapsed time for ground to reach during a pass, and the duration of contact outage. A summary of the variability in each of these parameters is given in Table 1.

4. *TM and TC for Daily Passes*

It was assumed that priority is given to communications over science, even when the pericenter occurs during a ground station pass, such that TM return is always 8 h complete in each pass. This was needed to satisfy the requirement for 500 Mbit/day data return.

Telecommanding was assumed as 4 h/day at least. With an uplink set to 1.6 kbps, this is a volume of 23 Mbits, or about 12,000 maximum size telecommands.

Table 1 Mars Express mission control parameters

Parameter	Minimum	Typical	Maximum	Dynamic range
TM volume return/day	500 Mbit/day	1000 Mbit/day	6000 Mbit/day	1:12
Pointing directions per day	1	6	20	1:20
Instrument ON duration per orbit	2 min	36 min	∞ (6.7 h full orbit)	1:270
Time to acknowledge 1 TC	10 min	20 min	44 min	1:4
Time to load and dump file (1000 TC)	~30 min	45 min	~60 min	1:2
Spacecraft elapsed times (observation → action) for ground to react during pass	12 min	22 min	46 min	1:4
No contact with Spacecraft	11 h	17 h	720 h	1:65

The daily telecommand contact is the baseline but is possibly not the optimal solution. For example, TM passes may be pre-programmed without commanding—taking into consideration longer-term events such as the station sharing with Rosetta during its Mars flyby, by which time the Beagle-2 mission would have ended so that short-loop telecommand cycles would no longer be needed. The MTL capacity and design of the spacecraft to survive unattended for 30 days suggested that contact every second day rather than daily would be feasible. It was assumed this could be tuned dependent on achieved autonomy and robustness. The future will show that the opposite solution has been retained, commanding several times daily rather than every two days.

E. Role of the Solid-State Mass Memory in the Mars Express Operations Concept

The functionality of the SSMM on Mars Express was always much broader than science data-storage and playback. The size of the MTL required and the limited processor memory capacity, as well as the need for secure and reliable file transfer of commands, gave the SSMM the status of "flying control center," in effect the last stop before spacecraft (in both directions). Ground commanding is essentially done via the SSMM. Deep space operations require safe uplink of a coherent set of telecommands, without waiting for acknowledgment of each telecommand in turn. This is implemented via the FT protocol. Files are transferred in small parts and merged onboard with a copy function. The data-management subsystem (DMS) software then uses the SSMM for telecommand handling: the SSMM supports automation via the MTL and TC files used in FT.

All telemetry packets generated onboard can be stored on selected SSMM Packet Store and can then be downlinked to ground via the TM Frame generator (TFG) (in VC1 frames). All packets can in parallel be downlinked to ground via the TFG, in VC0 frames. Normally all TM packets are stored in the SSMM; a few selected packets are also downlinked in real time. The SSMM continuously dumps housekeeping TM during a ground station pass. Any data stored in an empty packet store will immediately be dumped in virtual channel 1 (VC1) as soon as a dump is active. During a pass the SSMM is planned to be operated as a "bucket with a hole." All data were to be stored in a single packet store that is continuously being read out. This concept had to be changed later on.

Open issues at the time regarding telemetry included architecture — TM flow and optimization of the packet store definition; ground interaction — tradeoff between redundancy of real-time and replay telemetry and the lag time during a pass of visibility of housekeeping; ground-based handling (mission control routing of packets) and reaction to anomalies — time loss vs data loss.

Telecommands can originate from a variety of sources, including mission planning (payload requests and flight dynamics requests) and flight control team (procedures, tests). On the spacecraft the routing of telecommands can include the MTL, monitoring cells, and onboard command procedures (OBCP). The actual execution of telecommands can also be triggered by fault detection isolation and recovery (FDIR) and application programs (APs).

The MTL is stored in dedicated files within the SSMM and only an index table is stored and managed by the DMS processor itself. All MTL requirements are

handled by the DMS processor/software. The SSMM supports transfer of MTL in both directions (store and retrieve).

For the file transfer, within the DMS, the SSMM files are used to support requirements on the protocol between the ground and DMS. File transfer protocol requirements are essential for Mars Express to ensure mission integrity and autonomy. The requirement for storage capacity of telecommands in the mass memory is driven by the capability to store 48 h of commanding onboard in the MTL. The MTL is stored in files in the SSMM and is specified with a maximum capacity of 3000 telecommands. This is sized to cope with the various natures of commands from ground: attitude control, payload control, pass and downlink control, OBCP and software patching, and non-routine flight/contingency procedures for testing or commissioning. From the perspective of safety and security, the SSMM provides continuous memory scrubbing [error detection and correction (EDAC) protection].

IV. Mission Operations and Planning Concept 2002

A project-level review of the mission planning concept, drivers and constraints, tasks and interfaces, planned implementation and duration of cycles was held in April 2002. This was primarily to identify any shortcomings and risk areas and build confidence in the planning concept under implementation. This section highlights the changes in the operations concept since 2000 evident from this review and the reasons for these changes, including the realization of limitations, constraints, and open issues.

By this time the main actors of operations had been expanded to include the Lander Operations Center (LOC) at the Open University, for control of Beagle-2, a Project Science Team (PST), a Payload Operations Support (POS) Team, and ASTRIUM, the spacecraft manufacturer for long-term spacecraft support and consultancy. The expanded roles and interfaces are shown in Fig. 5.

A. Mars Express Operations Problem: A Variable Mission in a Variable Environment

1. Mission Planning Constraints

The mission planning concept is the embodiment of the operations concept, based on the nominal mission definition, and all of the constraints implicit within it. The constraints on mission planning derive from a number of sources, including the overall mission definition, operations principles, ESOC manning/effort, data generation and return, resource management and planning of phase-in and validation of the system.

As ESA's first mission to Mars, a cautious and safety-oriented approach to mission operations was mandatory, and operations of significant complexity were foreseen. This was the first “flexi-mission” with a different approach to the level of project-level review delegated to industry and the spacecraft manufacturer. The safety of the spacecraft has priority over the return of science data. The available spacecraft and ground resources, and science requirements, were to be optimized within the constraints defined for the mission without any risk to the mission.

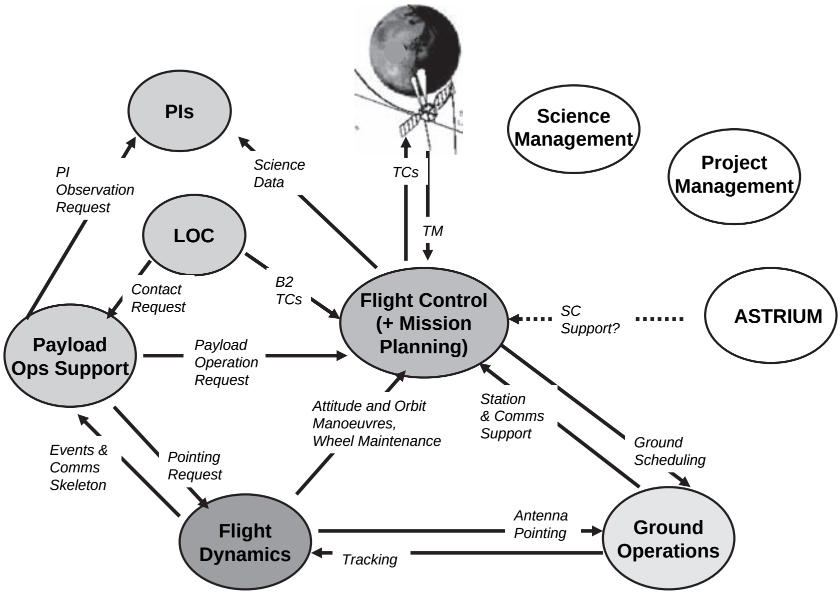


Fig. 5 Actors of operations (2002).

The mission planning concept is based on the nominal mission and the overall operations concept.

The science and spacecraft operations interact significantly. ESOC involvement in the mission planning concept started with analysis of the Science Timeline Analysis Tool (STAT) specifically for Mars Express. Validation of non-baseline activities and pointings evolved into identification of the "ESOC scenarios" from 2000 onwards. With no ROSETTA heritage to work on, a framework to the Mission Planning System (MPS) was developed in 2001. This same concept was also taken as the starting point by the Payload Operations Service (POS) at Rutherford Appleton Laboratory (RAL) in the United Kingdom in 2002.

In terms of operations principles to be applied to Mars Express, the basic constraints had hardly changed from 2000 to 2002. First, New Norcia is the prime MEX ground station, restricted to a single pass of 8 h per day within a communications opportunity defined by physical properties of the link. It is a shared resource with ROSETTA (and occasionally with Venus Express for radio science, and possibly later with Herschel-Planck missions). No real-time science observations (pointings) would take place during contact periods. A tradeoff between science and data-return/commanding is assumed with a granularity of one orbit. Quasi-real-time operations include telecommand uplink, health, and safety monitoring. During routine phase all spacecraft and payload commanding is via the MTL.

The limited financial resources have an impact on the ESOC manning for the mission that in turn restricts mission planning activities to normal working hours. The daily uplink of the commands to the MTL is to be performed by a shift

spacecraft controller team. Resources available to support the full planning cycle from PI request to final uplink are also restricted in the PI and project scientist teams.

Constraints relating to the generation of data and its return to Earth focused on the data-latency problem, namely that data should be dumped to ground within 24 h of their generation onboard, originally assumed feasible during the next contact pass. The principle reason was for ease of planning and to avoid building backlogs. Flexibility was provided within the planning cycle to optimize the plan for tradeoff between science observation and data return within the allocated science windows taking into account minimum window sizes.

Detailed management of resources had resulted in further constraints on planning. Within the coarse planning cycle, FD would allocate windows for communications and orbit maintenance, with pericenters reserved for science unless required for tracking purposes. Freezing of the spacecraft resources as early as possible would give the flight control and flight dynamics teams enough time to check constraint satisfaction. Request iteration was in principle excluded for cost and time reasons.

With regard to phasing in and validation of the MPS, this assumed that the planning concept applied to routine phase only, after successful commissioning of all payloads. The MPS would be phased in progressively and validated during Mars payload commissioning (to allow process refinement and optimization as mission stability is achieved).

2. *Mission Planning Assumptions*

In the case of the eccentric 6–7 h orbit chosen for Mars Express, the oblateness of Mars causes procession of the pericenter latitude, which completes a full rotation in latitude from pole to pole again in about 11 months. This allows full coverage of Mars for mapping about twice during the nominal mission of 23 months (one Martian year). The frozen orbit concept is used to maintain orbit as close as possible to a pre-defined reference orbit with no uncontrolled drift. This allows prediction of the orbit and its associated events as early as possible and pre-planning of science and spacecraft operations, evolving from course to more detailed planning with advancing proximity to execution.

A Master Science Plan generated under the management of the project scientist was assumed to define the priorities, sequences, and phasing of the science mission. It would involve all parties including PIs, payload scientists, POS, FD, FCT, and the Mission Planning Team. This plan was intended to have the granularity of *one* orbit.

Management of changes in the planning was assumed to involve iteration cycles shown in Fig. 6 [1,2]. The PI and POS teams would compile the science and pointing requests (PTR) file. POS and FD would arrive at a stable timeline to fix the pointing requirements. The POS and MPS teams would then arrive at a consolidated detailed schedule for payload operations.

Late changes, especially those involving changes to resource allocation, would have required a complete reconsolidation of the plan (full iteration) and were in principle excluded. For example, moving a radio science bistatic radar (BSR) request due to ground station availability would change the power, data downlink (during BSR the spacecraft uses inertial pointing at the planet and cannot downlink

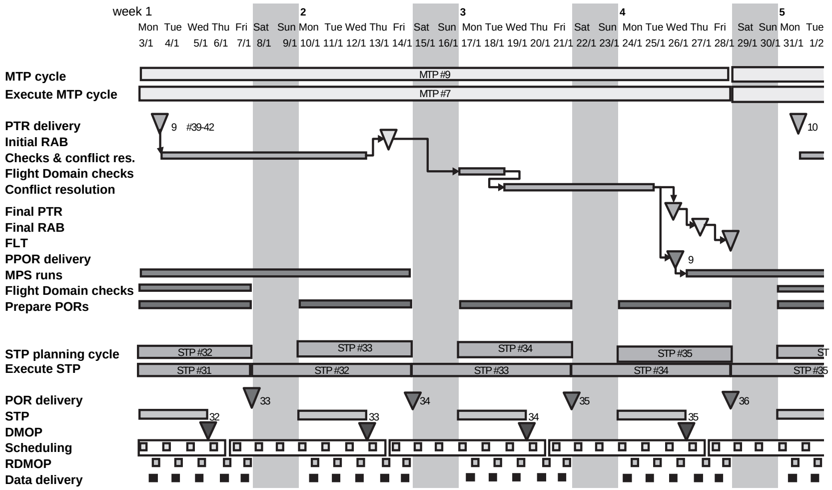


Fig. 6 Mission planning timeline concept.

data to ground), radiator illumination accumulated flux, etc., leading to a full iteration of the PTR. Telecommands for a minimum of two days (6–7 orbits) must be held in the MTL buffer onboard. The commanding plan shall always allow for complete loss of one ground station pass.

Payload operations requests (PORs) are assumed to be defined to allow discrete planning on an orbit-by-orbit basis. It is assumed that all POR contain a single operations request, i.e., only one POR per orbit per instrument. This allows introduction of late modifications with the restriction that there is no increase in demand for resources. Such single operations requests can be suppressed without any complete replanning cycle, since they can only release resources, and cannot impact on other planned observations.

Planning relative to events (such as pericenter or apocenter time) was introduced. The frozen orbit with its predictive events allows execution times to be requested relative to any event. Absolute times are not exact enough until the orbit has been accurately determined. Use of the latest orbit data available is ensured to achieve the accuracy required for instrument and pointing timing, while meeting the requirement for early delivery of operations requests to allow the mission planning team to have sufficient time to check constraints.

In early phases of planning, an uncertainty of 2 min needs to be taken into account for orbital and spacecraft events such as pericenter passage, momentum wheel off-loading, maneuvers, etc. Absolute times of command execution are then computed by the Scheduler Function in the MPS based on the latest available event file at time of final command generation.

3. Mission Planning Drivers

The drivers for mission planning concept and timescales derive principally from the mission implementation, the spacecraft design, ground considerations, and operations principles.

With regard to the mission design, all operations are performed “out of contact,” either with a one-way delay of between 4 and 22 min or outside station coverage. Spacecraft safety is paramount and operations shall never trigger entry into safe mode. This requires that operations are pre-planned and pre-checked for safety [3]. The changing environment and key variables (power, thermal environment, data rates, etc.) imply that experience gained requires extrapolation and therefore extra margin when preparing future “similar” periods. Pre-planning and checks taking into account of the specific environment before uplink are essential.

Flexibility with respect to the design case has cost implications. Extra checks are required, and it is more difficult to find acceptable solutions in case of conflict. Extension beyond design case requires more advanced planning (and a higher risk of failure). The orbit is frozen; ground track and environment conditions are known and fixed, so that precise operations can be planned months in advance. However, confirmation of their feasibility in a given environment (thermal, eclipse, etc.) will only be acquired progressively with experience.

From the spacecraft perspective, all operations would be store-forward in practice. The entire schedule is conceived to be uplinked each day for two days in advance. Slews and wheel off-loadings (WOLs) cannot be calculated many days in advance because of orbital perturbations and variations. The pointing strategy must be fixed and known to be solvable before entering the short-term planning (STP) cycle. The spacecraft pointing defines the activity and many resources (power, thermal, illumination fluxes, etc.). A feasible strategy must be agreed on before final resource checking, to minimize the iterations.

Ground considerations require many activities to be run in parallel to prepare for (i.e., plan), execute, and report on operational activities, implying long lead times on inputs. As the schedule is generated, each day plan is finalized (passes all checks) one week in advance for the next week’s activities (STP cycle). The pointing also defines communications, thus pointing definitions are needed months (MTP cycle) in advance for ground station scheduling (and release). The mission is resource limited, sized for coping with the “design case” mission and flying “within the box.” Either resources must be increased (a bigger box) or sufficient time for all checks must be allowed in advance (to avoid “flying outside the box”), i.e., in conditions not validated at least at planning level.

4. *Compatibility of Science Scenarios with the Spacecraft Capabilities*

During the spacecraft Critical Design Review (CDR), concerns were raised about the approach used by the project team to check science scenarios with respect to (design case) spacecraft capabilities. The action was passed to ESOC to simulate the performance of the spacecraft, including flight dynamics, thermal, and power resources, in order to demonstrate the capability of the Science Master Plan, once it was defined by the Science Working Team (SWT) for the first 6 months of operational mission.

At the time the Master Science Plan was not available, and so there were no scenarios to check against “presumed” capabilities.

The mission simulation process, described in Table 2, is designed to evaluate whether the spacecraft can carry out the plan defined by the MSP is not classically part of the mission planning process.

Table 2 Planning processes — mission simulation

	Mission planning	Planning simulator	Mission simulation	Spacecraft simulator
Function	Plan (and schedule) science (and control) Spacecraft (and ground) activities	Verify feasibility of set of activities for a planning period	Verify feasibility of mission profiles (mission analysis, flight dynamics, flight control)	Test procedures, train the team
System	MPS	MPS modules	Process: NO SYSTEM	SIM
User requirements document	MPS concept → MPS URD	- OCD models (power, data) - ASTRIUM model (thermal simulator)	Extended MPS concept (including flight domain checks at MTP level)	SRD
Specified by	FCT (MPS engineers+)	FCT (MPS engineers+)	So far: FCT (overall), ASTRIUM (thermal)	Sim Technical Officer + FCT
Developed by	Contractor (Anite)	FCT (MPS engineers+)	To be discussed	Contractor (Vega)
Heritage	Envisat	Approach: ERS, Envisat Rules: MEX specific	NONE (new approach at ESOC for ROUTINE)	ROSETTA, XMM
Validated by	MPS Technical Officer + FCT(MP)	ASTRIUM? ESTEC experts?	??? (ASTRIUM, mission analysis?)	Sim TO + FCT
Used by	FCT (MPS engineers+)	MPS engineers	To be set up	FCT, Sims Officer
Timespan of usage	(end of commissioning), ROUTINE	ROUTINE only	Mission preparation and routine	Sims campaigns, all non-routine activities

If the mission simulation capability could not be demonstrated, what alternative course of action existed was therefore not easy to derive from ESOC's extensive flight operations experience. Whether this could be done as "normal work" with no extra resources was also questioned at the time.

This "mission validation deadlock" was to be solved by a much more interactive approach, combining learning "on the fly" and a complex but coherent and strictly controlled mission planning process as described next.

B. Mars Express Operations Answer: Planning Gives the Flexibility to the Operations Concept

Basic mission principles must be critically reviewed to understand how flexibility within the operations concept can be exploited. Many of the adaptations to the operations concept later applied to cope with changes in the baseline and contingencies were made available from this inherent flexibility.

The spacecraft has been designed for a certain sizing case including reference operations scenarios or cases. However, more than reference operations cases are supported by POS/ESOC and the spacecraft under certain conditions — the problem being to quantify what is meant by "more." It is assumed that extensions will be introduced progressively during real operations at Mars.

Spacecraft support activities are only conducted for the purpose of the science mission, and these increase in proportion to science activities (e.g., wheel offloading), implying a tradeoff: the more the spacecraft is used outside the reference case, the more this must be planned ahead (i.e., longer planning cycles).

The spacecraft executes a defined mission within cost and schedule constraints. Its capabilities are limited in many respects, for example, power output fixed by solar array area, battery storage capacity (and aging), wheels size/performance, and fuel mass available.

The need for full spacecraft re-pointing for science observation vs communications (Earth pointing) is the major driver for optimization of these resources. It almost always fulfills a given operations reference case, as defined in the Mission Reference Scenario: the lower parts of the orbit are primarily used for observations and Lander communications, with the instruments line of sight pointing nadir. The upper part of each orbit is used for communications, with ground station, HGA pointing to Earth, when visibility allows and for battery re-charging. In addition, special science operations, requiring non-nadir or higher altitudes, can be scheduled (resources permitting). The hardware sizing case guarantees science operations on all orbits (although this could be prevented by some conflicting inevitably with communications windows, depending on the priority retained for this orbit).

This flexibility from planning allows here a *quantum leap* to be made; the granularity of one orbit does not have a resolution good enough; the orbit needs to be split into subphases distinguishing observation, communications, and other spacecraft activities such as orbit maintenance. This decisive change in the concept needs to be supported by a specific detailed implementation described next.

The concept offered to the PIs, *in addition to the baseline mission*, includes data return optimization, a larger choice of targets via selection of versatile pointing for observations in any nonreserved window, extended observation windows

(nadir above 1200 km altitude), late selection of instrument parameters (to within 1–2 weeks of execution), and improved timing accuracy of operations execution (using latest available event data). It does not provide improvements outside the capabilities of the spacecraft. It does not allow permanent erosion of margins or drift in resource balances due to occasional operation outside of spacecraft limits, nor for late changes to pointing selection.

1. *Flexibility A: Observation vs Communications*

The first flexibility provided by planning is the ability to give priority to science or to communications. Long-term planning of communication and data start from the flight dynamics long-term event file (EVTF), long-term orbit and reserved windows — the flight events and communications skeleton (FECS). Spacecraft maintenance and communications windows are chosen to preserve as far as possible the pericenters. A lesson learned was that the spacecraft (orbit) maintenance could be phased with the orbit (at apocenter) preserving the pericenter for science, but not ground station time (phased with Earth rotation). The only way for the operator (ESOC) to reserve pericenter-free communications time every day was to reduce the dump time or extend the ground station coverage, as one will see later.

PIs and POS together will maximize science data return (through tradeoff of observation with communications). On each day a choice must be made between 6.5 h of data downlink with all pericenters taken for science, or 8 h of data and two pericenters out of three available for science. Another lesson was that this assumed full stability of station availability timings — not always true for a large virtual network [also including NASA Deep Space Network (DSN) support]. The output is the science events and communications file (SECS) representing a frozen communication baseline. ESOC follow-on is to allocate all relevant ground resources.

2. *Flexibility B: Pointing Direction*

The second flexibility offered is in pointing direction. Medium-term planning (monthly) defines the pointing direction. Starting from the Master Science Plan, PIs and POS construct and pre-check pointing request. Tradeoff observation subjects depending on mission phase (normal coverage, Lander support, occultation season, etc.). FD checks the pointing request (PTR) and elaborate attitude timeline to generate a frozen and feasible pointing timeline (FTL). ESOC follow-on is to prepare maneuvers for execution.

3. *Flexibility C: Observation Duration*

Extended observation duration is the third flexibility provided by planning. The allocation of spacecraft resources in the MTP cycle, shown in Fig. 7, again starts from the Master Science Plan. PIs and POS construct and pre-check observations slots per instrument [with the MEX Instrument Resource Analyzer (MIRA), a POS facility derived from the original STAT tool from 2000], possibly extending the nadir. A tradeoff (linked with pointing) is made for data share,

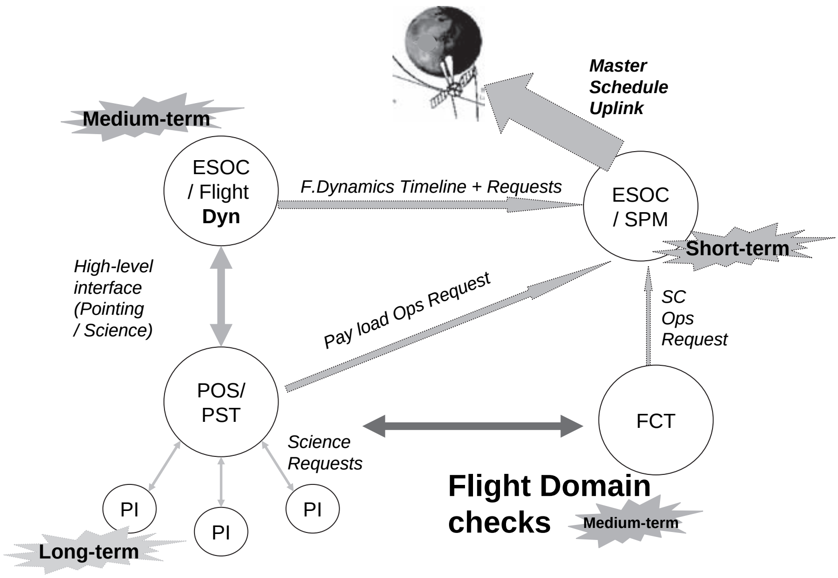


Fig. 7 Functional overview of mission planning cycles.

modes, instrument internal timeline, etc. ESOC Mission Planning Team and FCT perform flight domain checks (PTR consistency, power, thermal, operations resources, etc.). The output is a medium-term observation plan for the month after next. Allocation of spacecraft resources and preparation of spacecraft operations requests (SOR) is an ESOC follow-on activity. The SORs cover transmitting times, dump plan, attitude and orbit control subsystem (AOCS), operations, etc.

4. *Flexibility D: Instrument Tuning*

Late selection of instrument parameters is the fourth flexibility provided. Within the short-term planning cycle, integration of actual requests is made. Starting from the flight dynamics timeline and initial PORs (known as POR lites), PIs and POS provide final PORs making the tradeoff for instrument internal settings (not impacting spacecraft resources). ESOC/MPS perform all mission planning checks and output the Detailed Mission Operations Plan (DMOP), which must be both feasible and committing for the following week. Scheduling (of ground facilities) is an ESOC follow-on activity.

5. *Flexibility E: Time Accuracy*

The final flexibility provided is execution time relative to defined events. Spacecraft programming is a very short-term or scheduling activity. Starting from the DMOP, wheel maintenance and maneuver parameters provided by FD with SSMM handling and data recovery from FCT are used to construct slices of the

master schedule covering two days of the mission. ESOC are then responsible for its uplink each day during contact.

These five fundamental degrees of flexibility have been fully provided in the real implemented mission.

C. Flight Dynamics Operations Concept

1. Long-Term Flight Dynamics Planning

The concept for flight dynamics operations is based on the definition of a frozen ground track. The strategy is defined by required spacing at the ascending node to achieve no gap and minimum overlap between orbits to define the nominal ground track—optimized for global mapping with the stereo camera. Maneuvers at apocenter are designed to keep frozen ground track by correction of accumulated deviation. Initial estimates budgeted for one wheel de-saturation (WOL) per day, with an orbit correction maneuver (OCM) when the offset between nominal and real ground track is above a prescribed threshold. The WOL is used for orbit maintenance, which implies proper positioning of desaturations. The windows are defined by FD and must be between 90–270 deg true anomaly to preserve each observation arc. The selection of the orbit must satisfy all requirements: global surface coverage from low altitudes with good illumination conditions for optical instruments.

Tracking requirements are a fundamental driver for the FD concept. Tracking data are used for orbit determination. For good orbit accuracy it is required to get data from all parts of the orbit, but the spacecraft must be Earth pointing while taking tracking data, which is in conflict with science observation. Ground station availability will overlap with roughly one in three pericenters. The science observations during this revolution shall be limited to a minimum observation period of 1.5 h including slews. Outside this period the spacecraft shall be Earth pointing, thus allowing for communications and tracking data.

Over the long-term six-month timescale, FD provides the baseline reference orbit and event files, used throughout the complete mission planning system and cycles. FD must guarantee that the actual orbit is close enough to the baseline (within defined constraints, for example, 2-min accuracy on pericenter time). Slots are reserved within the FECS for orbit maintenance: WOL and orbit correction maneuvers. The selection is subject to complex analysis of likely orbit perturbations for each mission phase. WOL are coupled with orbit control, the ΔV effect of each WOL is strongly influenced by the position in the orbit and the spacecraft attitude in which it is executed. It shows that in principle orbit control can be achieved, but a tradeoff with operational simplicity shall be considered. Ideally all WOL slots would be at apocenter in Earth pointing attitude. Initially one slot per orbit for WOL was foreseen and one per week for orbit control.

ESA had at this time only a single deep-space 35-m antenna at New Norcia (NNO) in western Australia. This must support the ROSETTA comet mission in parallel. Some ROSETTA critical phases overlap with MEX routine at Mars. MEX anticipated a full 8-h allocation on a daily basis except where pre-empted by ROSETTA priority pass. Planning will seek to optimize the total visibility time that can be granted to both ROSETTA and MEX over various phases.

The ground station time allocation process is as follows. FD generate the FECS using pass estimates generated by the Station scheduling office. Products depend on known requirements from all missions sharing the ground station, available visibility predicts, and resource allocation rules/guidelines (not yet defined in 2002).

Sharing rules would be approved and refined at a higher management level comparable to NASA Resource Allocation Board for DSN, a proposed short-term approach based on initial support requirements, agreed/tuned with ROSETTA. The POS can progress with MIRA for initial science planning before availability of the first FECS.

2. *Medium-Term Flight Dynamics Planning*

The POS provides pointing requests to FD in the form of the pointing timeline (PTR) that covers four weeks of operations (100 orbits). FD then finally confirms that the pointing requests are dynamically feasible. FD defines a flight dynamics timeline (FTL), which in essence describes the pointing sequence including pointing requests from the POS, attitude slews, special attitudes for WOL, orbit maneuvers, and Earth pointing phases.

The medium-term plan (MTP) was not foreseen as an iterative process (although in routine operations it unavoidably often is). The pointing requests generated by the POS shall take into account all existing constraints. It is foreseen that the POS fixes pointings for one month of operations, between one and two months prior to the operations execution. Shorter planning cycles (covering, e.g., one week) were assessed and discarded, because they would increase the interaction between POS/FD, for which resources were not available. Shortening the processing period would increase the risk of cancellation (as opposed to re-optimization). All information required for planning the pointing strategy will be available several months in advance.

The allocation of slew times (durations) has significant impact on the operations concept. Mars Express is not an extremely agile spacecraft from the dynamics perspective. The spacecraft design scenario covers nadir around pericenter and Earth pointing elsewhere. It can support more but with limitations. Reaction wheel size (12 nms) is relatively small. These are used both to compensate for disturbance torques and to provide control torques. Constraints exist on reaction wheel zero crossing. Relatively long tranquilization times after each slew, of up to 15 min, result from the long (20 m) and flexible Mars Advanced Radar for Subsurface and Ionospheric Sounding (MARSIS) radar antennas. It is the task of the POS to generate pointing requests compatible with spacecraft resources in terms of slewing capability. ESOC/FD provides for checking slew time feasibility to the POS. ESOC/FD would check the feasibility of attitude slews between two consecutive pointing profiles. Attitude slews for MEX are inherited from ROSETTA. The tool provided by FD for slew checking is, for those parameters that cannot be anticipated in advance, based on budgets. It results in worst-case computations. During actual operations slews may in general be shorter, sometimes much shorter.

Off-track pointings were considered feasible for MEX, and had been validated at the spacecraft design level. These come with certain impacts on power budget, thermal (solar illumination) aspects and possible star tracker blinding by the sun.

They also have an impact on wheel loading (prediction and execution). Off-track passes destroy the contiguous global Mars coverage (for mapping). The nominal orbit/control strategy is aimed to provide good coverage for nadir pointing. The off-track affects essential spacecraft resources and cannot be decided on a short basis as it implies a different loading of the wheels. The slews planned afterwards may then become invalid.

Also possible was nadir with a fixed offset of less than 30 deg—the so-called across-track pointing. Again power, wheels, and illumination restrictions limit this in practice. It replaces the nominal nadir (around pericenter), provided that the planning impact is acceptable. Inertial pointings are restricted by slew maneuvers (duration, wheel profile) and solar illumination on some surfaces (cryogenic radiators, instruments, and thermal radiators). Possible extensions to the available pointing types for science were considered. The baseline was nadir pointing for the full period of 36 min during which the altitude was less than 1200 km. This constituted the nominal mission and was always possible. Extended nadir below 4000-km altitude is limited principally by power, but was considered feasible outside of communications pass and outside eclipse seasons.

D. Planned Implementation of the Retained Concept

1. Recommendations for Science Planning

Before any experience could be gained at Mars, several general recommendations were made by ESOC on science planning to initiate the process and ensure the feasibility of the operations concept. Instead of the baseline "one pointing per orbit," science pointings would now be limited to one nadir and one inertial (or maximum two) science pointings. It was not planned to perform nadir and off-nadir around the same pericenter. An average over planning period should be between one and two science pointing requirements per orbit. The separation between two pointing types should be at least 30 min, whatever the angles of the slew. Between any two science pointings Earth attitude should be restored. No extended nadir pointings would be allowed in eclipse season. As yet unknown thermal constraints (due to solar illumination flux) would have to be considered. A minimized number of interruptions during pass (for science) is required to ensure data return. Consider trading off pericenter observations in station visibility for mission phases with bit rates below 30 kbps (500 Mbit total science return required for a single ground station 8-h pass).

The return to Earth pointing attitude between science pointings was non-negotiable and implicit in AOCS simulations during design, and procedures/command sequences generated from the Spacecraft User Manual. The 30-min separation between pointings was calculated from a typical Earth-nadir or nadir-Earth slew duration (15 min average each at maximum wheel speed, along direct path-arc of great circle). Worst-case slew study was ~50-min duration.

To stay well within the power limitations, no extended nadir in eclipse season should be planned. The exception is for seasons where the orbit plane (nearly) orthogonal to sun direction, during which full power on the arrays can be extracted throughout nadir pointing, shown in Fig. 8: from power point of view even the full orbit could be nadir!

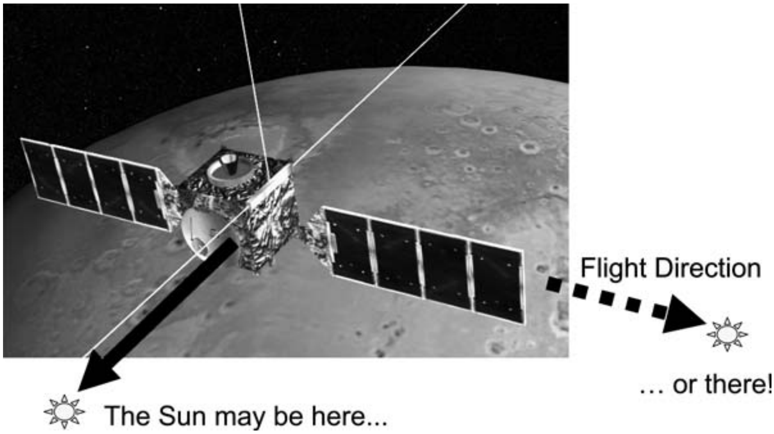


Fig. 8 Availability of power during nadir pointing.

During eclipse seasons the orbit plane is roughly parallel to sun direction and a nadir pointing could discharge about 15–20% of the full battery capacity (at beginning of life). Extended nadir pointing could cost as much as 60% depth of discharge (DOD), which is not acceptable operationally. The acceptability of extending the nadir depends on several considerations. First, for the initial conditions (minimum state of charge of batteries), at eclipse entry or nadir the DOD must be less than 10%. This also implies no overlap between the eclipse and nadir phase (which could cost another 10–20% DOD). Second is how much power can be collected during the nadir (i.e., the actual angle of the orbit to the sun direction). Third, a margin against further discharging activities such as safe mode must be considered (at least 10% DOD). Finally, the recharge subsequent to the nadir pointing must be sufficient to achieve minimum requirements at the next discharge event. Battery aging (lithium-ion cells) after several years in flight is not well known and must be monitored by ESOC. The primary goal is to avoid safe mode and mission interruption—recovery could cost two to three days or about 15–20 pericenters lost to science.

Thermal rules were not defined at the time. It was recognized that rules like "So much flux allowed for that long on that face" (e.g., payload face) would be required. These were already needed for deducing "simple" planning rules for the POS/PIs.

Baseline data downlink is from cyclic packet stores (start/stop without pointer nor file handling by the ground), which allows as many interruptions as needed (uplink sweep, pericenter, occultation). Because of recognized performance limitations in the SSMM, some packet stores (for all small size packets) would have to be dumped via a mechanism known as Plain File Reports (PFR) [1]. The PFRs required use dump commands that are not interruptible. This introduced a requirement to minimize the number of interruptions in any pass. Some breaks were unavoidable, for example occultations of the Earth by Mars itself. These SSMM limitations for daily control and data handling impact the flexibility of the science planning.

At the time data shares between instruments are ensured by daily dump of the data acquired in the previous three to four orbits (24 h) and immediate cleaning of the file. A “flip-flop” mechanism between two identical files in the SSMM allows recovery beyond a single day if there is margin in the downlink. This implied that the Master Science Plan would define a daily data share per instrument (to generate the SSMM start/stop). Since packet stores have to be dumped fully, any suspension of the ground contact (including for pericenter) will introduce some inefficiency in the overall dump sequence.

2. *Phasing in the Operations*

A progressive extension of operations was assumed under certain conditions. The First Flight Rule is: Spacecraft safety always takes first priority! Second, spacecraft basic operations at Mars are secured. Third, additions and extensions are introduced progressively. Finally, the combination of mission conditions, spacecraft status, and request for the planned operations is validated for the planned execution date (using available tools and expertise).

A typical schedule at Mars was foreseen with commissioning for two months after arrival on final orbit (including trial/validation of non-baseline activities). From +2 to +6 months after final orbit baseline operations for routine phase would be proven, including the first eclipse season. After +6 months non-baseline activities would be included in routine mission planning.

The implementation of the operations concept until Mars arrival would focus on several aspects. First priorities are readiness for launch and to reach Mars orbit safely. Second is implementation of the mission planning as defined previously. Then the integration into MPS of “flight domain” checks to complement and coordinate with FD. This required the “simplified” thermal/power model from industry (known as the mission simulator), the specification of which was ongoing. The MPS would be configured according to variable resource profile of each planning period. Pre-validation runs are to be performed on consistent Master Science Plan segments, when available, followed by actual validation during mission operations.

Expected from the POS and science community was a top-down Master Science Plan covering the whole mission, including detail of science pointing and instrument complement for each and every orbit over the first six months (commissioning + initial routine). This would be ready to run in parallel to Mars commissioning. A re-plan of the initial four months of the routine phase could be required by actual Mars injection. Also prepared in advance would be the second long-term cycle (another six months).

3. *Planned Improvements*

Four planning phases were envisaged, 1) Mars commissioning, 2) consolidation, 3) routine phase A (planned before Mars arrival), and 4) further routine phase. For each phase different observation profiles would be allowed. For commissioning one pointing per orbit (nadir–normal, offset or extended, Lander nadir or inertial), defined by the commissioning plan with instruments defined by the SWT was

foreseen. As phases 2, 3, and 4 progressed, defined by the Master Science Plan, number and complexity of pointings would progressively increase.

Improvement of the planning cycles at all levels was foreseen. At short-term planning (STP), the high workload of planning engineers, daily scheduling is time constrained but can be optimized with additional resources and automation of ground systems. At the medium-term planning level performance increase was anticipated with time gaining experience, enhancing and optimizing work flow during the consolidation phase (after Mars commissioning). Compression of a cycle was only to be considered after the consolidation phase in agreement with POS, FD and Mission Planning.

Improvements of operational capabilities potentially included tracking (reducing the constraints on communications windows), payload operations (allowing start of some operational payload modes before nominal tranquillization time), pointings (back-to-back pointings with no return to Earth attitude in between), and power (minimizing the operating margin during low power Mars aphelion season) and data (refined low-rate TM modes and optimizing SSMM packet store assignments for payloads).

V. Lessons Learned Through Evolution of the Mars Express Operations Concept

This section highlights the evolution of significant aspects of the operations concept from 2002 until arrival on the final Mars orbit in 2004, and since start of routine phase operations. First, the open issues not resolved at the time of the operations concept are described and their implications for the concept. The changes in the mission design since flight commenced are then outlined and each is then discussed for its impact on the operations concept.

A. Need for Science Planning Tools

A requirement emerged for tools to allow the science community to assess long-term science planning at granularity below orbit level. Selection of pericenters for science (a tradeoff with communications) would have to consider many constraints and resources to reach an optimum solution, the optimum varying depending on the considered timescale (short, medium, or long term). Many conflicts between different instruments were anticipated [pointing, power, onboard data handling (OBDH) bus usage, data generation volume, etc.], but no practical way of resolving these existed at the beginning.

Already in 2002 the introduced “margins” had a magnitude well beyond the nominal baseline in most respects. Adding margins for operational capabilities with factors of two to three times the nominal capability was already opening the definition of a completely new mission baseline.

This has included a change in status of long-term planning from a *declaration* of resource allocation and priorities over a 3–6 month period into a *wish-list* regarding preferences for resources and utilization that is subject to late changes and events outside the mission itself. The long-term plan was supposed to be a declaration of station allocation six months in advance. Being part of a virtual

network offers access to many more stations from ESTRACK and DSN, but implies being influenced by far more variables also. The capability to react to constraints imposed by other missions is of paramount importance and is offered in the case of Mars Express by exploiting the operations concept flexibilities.

B. Planning Flexibilities Become the Mission Baseline

The flexibilities intended to extend mission opportunities described in Sec. IV.B. were originally conceived in 2002 as potential ways to increase the science return during routine phase. Planned introduction one-by-one to be phased in after commissioning was foreseen. With flight experience, improved MPS modeling and planning, and better understanding of the mission environment, these rapidly became routine mission baseline modes and operations. For example, some months of the mission are flown almost entirely without a single nominal nadir pointing—originally the mission baseline—using exclusively inertial or across-track pointings. Further pointing types have progressively been introduced well beyond those foreseen in 2002, including spot-pointing, specular pointings for bistatic radar observations, along-track observations, and nadirs that evolve into an across- or along-track pointing.

The change of paradigm—from a rigid operations concept to one that is flexible with rigidity when necessary—has occurred over the duration of the mission. This has meant progressively taking flexibilities in the operations concept and turning them into the mission baseline for operations.

C. Power Subsystem Performance

The detection of an anomaly in the wiring between solar arrays and the array power regulators (APR) on Mars Express was only made in-flight during near-Earth verification of the spacecraft performance. This fault allowed only 72% of solar array power to be extracted from the power subsystem. This of course could not be planned for in the operations concept before flight. It has since become a major driver for spacecraft monitoring, in particular on power and thermal control subsystems and also for the modeling fidelity required in the planning systems (MPS). The history of the power subsystem and its modeling, however, the refinement of the thermal subsystem power forecasting highlights the importance to the operations concept [4].

D. Virtual Network of Ground Stations

Although identified as an area of concern, the strategy to resolve conflicts on the demand for pass time over the single New Norcia ground station between ROSETTA and Mars Express had not been fully concluded. Changes in the allocation of passes between missions, for example during ROSETTA flyby of Mars, would impact the pericenters available for routine science, but the Master Science Plan would not initially cover this period. A transition eventually materialized after 2002 from a concept of “sharing a single ground-station” to becoming a part of a large, multi-agency virtual network of deep space antennas.

The initial concept was based on the shared use of a single ESA deep space 35-m antenna at New Norcia. Final agreement in 2003 was reached on the allocation of shared passes over 34 and 70 m from the NASA Deep Space Network (DSN) antenna. This would contribute to the data return from the MARSIS (ASI/JPL contributed) radar and on the 70-m dishes for bistatic radar experiments. Pre-allocation of DSN passes, typically for 8–12 h shared between two stations (Madrid and Goldstone) each day, allowed for a total communications window of 12–16 h each day split into as many as five mini-passes. Many of the DSN 34-m antenna use multiple spacecraft per aperture (MSPA) technology to provide downlink from up to four Mars orbiting spacecraft combined. Passes can be taken on any of 13 different DSN antennas (excluding 26-m dishes). This complex ground station network can be envisaged as a "virtual network" that must be planned and optimized for Mars Express, at each level of the planning cycle (long-term, monthly, and weekly). This virtual network has significant implications for the complexity of data downlink control, station "keywords" generation (for automated configuration of the station), and spacecraft controller manning levels.

With the availability of an extensive and flexible virtual network of ground stations, it was possible to develop rules to prioritize selection of the ground station where allocation overlapped between two antennas. First priority was always given to 70-m-antenna coverage, where this was allocated. This would maximize data return during low data rate seasons. Passes were split between those with and without uplink (MSPA downlink only). Uplink duration and distribution became a new resource to optimize. Passes with uplink have priority over those without. Finally, a signal is always kept from a station, and switch to another station is normally delayed until the first allocation is over. These concepts when combined together result in an operations field new to ESOC, well beyond the extension of the "sharing a single antenna with Rosetta" concept originally conceived in 2000.

E. Heater Power and Solar Illumination Constraint Modeling

With an installed power of over 500 W and a typical consumption of 80–250 W, the thermal control subsystem represents the largest single contributor to the power consumption on Mars Express. Following the restriction of power availability to 72% of solar array capability, the necessity to model the heater power consumption within the planning system became a priority. This was already initially foreseen in 2002 with the provision of the ASTRIUM power and thermal simulator to predict a heater power demand profile for use within the planning system. However, validation of the simulator against flight data was never achieved, in part because small errors in the prediction of temperatures in the thermal model would lead to significant errors in the heater power demand controlled by thermostats and software measurement of thermistors. An empirical model was developed based solely on historical power demand data and the known environmental variables (solar constant at Mars and changing sun-spacecraft-Mars angle), to predict the average heater power demand (outside eclipses). A model of the additional heater power demanded during eclipses was constructed from TM data alone. These together provided a model of heater power demand that was accurate to ± 10 W on average over a day.

The spacecraft manufacturer provided an assessment of allowed solar illumination angles and duration. These were based on the design case and a typical 36-min baseline nadir. These "illumination rules" included restrictions on the spacecraft primary radiators (±*Y* face) and the instrument mounting face (+*Z*). Based on flight experience over the first months of routine and commissioning (and predictive sun angles files), these could be evaluated as an actual accumulated flux, and assessment of temperature impact of various illuminations was evaluated. These actual flux values varied over the first Martian year. Maximum "flown" values could then be used to set numerical limits (as opposed to angle-time restrictions) and these were then applied in the planning process at the MIRA level. In this way the natural variability of an eccentric-orbit Mars mission was exploited to provide the flexibility in pointing required by science.

No rules on the solar illumination of various surfaces with restrictive temperature requirements had been explicitly defined in 2002. These could be extrapolated for some surfaces implicit in the design case of a 36-min nadir even during Mars perihelion outside eclipse season (e.g., main spacecraft radiators). For others, such as the cryogenic instrument radiator, no solar flux illumination was foreseen during pericenters, and other ways to elaborate rules had to be found later. The contribution of planetary albedo and Mars-shine was not considered significant at the time.

F. Data Downlink Prioritization and Automation

The added complexity of the ground segment operations, or virtual network, with its corresponding change to five to seven mini-passes each day, and the impact of limitations in the SSMM, resulted in significant complexity in organizing and optimizing the dump commanding of science data in the MTL onboard. Attempts to automate this programming culminated in the development of the MEXAR data dumping software system based on artificial intelligence techniques of constraint resolution [2].

G. Routine to Critical Operations Mission Phase Transition

After the completion of commissioning at Mars, the transition between routine phase operations and non-routine or critical phase operations was not intended to occur in-flight with any regularity. The delay in deployment of MARSIS radar dipole antennas, until long after entry into routine phase operations on the other instruments, an average of three safe mode entries per year in 2004 and 2005, and several changes in the orbit have made necessary such a slicing of the "routine" phase by non-routine periods of activities. Resumption of the science mission can now be made within 24 h of safe mode exit. This is largely due to the flexibility provided by the planning concept and the ability to pre-define a medium-term plan that can be entered or departed at virtually any orbit boundary.

The regular need for spacecraft-level testing and commissioning of new modes was also not planned. For example, the development of a survival mode (SUMO) for the aphelion-eclipse season in autumn 2006 requires extensive platform reconfiguration, and testing in parallel to routine phase operations is necessary.

The capability to perform routine phase observations in parallel with background commissioning was not foreseen but has been proven in-flight on many occasions.

This capability is now frequently used to further enhance the mission or to refine operational configurations to better meet the needs of changes in the environment, without any halt to the routine science mission. For example, the testing of power and thermal configuration changes to minimize power requirements during the Mars aphelion-eclipse season in late 2006 has been ongoing for several months. The ability to perform these on a resource-limited spacecraft has evolved from the operations concept defined by flexibility with occasional rigidity.

VI. Conclusion

This work has focused on the history of a single planetary science mission and the evolution of its operations concept. Lessons can be learned on what must be known, and by when, to derive a stable concept and in particular to understand its in-built flexibilities and margins inherent in the operations concept and mission design. Examination of these flexibilities can reveal potential extensions in the capabilities of the mission to achieve its objectives, or to overcome any shortcomings experienced in flight.

The most important conclusions that can be made from this study are summarized as follows:

1) Plan science with tools, if not with rules—mission planning tools can be used to evaluate the resources (such as power, data storage, and downlink) and constraints (e.g., solar illumination flux) resulting in a safe, achievable science plan. The tools are based on empirical or engineering design models within the planning system. Often most of the constraint violations can be eliminated with simple up-front rules, such as "only two pointings per orbit allowed."

2) The operations concept shall be at least as flexible as the mission definition: a Mars orbiter mission by definition includes system margins in terms of power, data, and other sizing parameters to cope with the varying solar constant at Mars, varying Earth-Mars distance, and signal return time. The operations concept must take into account the flexibilities provided by these system margins.

3) Flexibility saves the mission return in case of major mission anomaly (on the spacecraft, e.g., power, in planning, e.g., late DSN station reallocation, or one-off emergency, e.g., Mars Global Surveyor recovery search). If the operations concept already has in-built all the flexibilities provided by the mission design margins, then even a major anomaly that removes all the apparent system margin can still be accommodated.

4) Integrate all ground resources in a virtual network, even when (and in particular if) they do not belong to you. A planetary orbiter is fundamentally constrained by the ground resources available. If they are not collectively considered as a virtual network, then changes in the resource baseline will have severe impacts on the science return.

5) A complex mission simulator may to some extent be replaced by an empirical, but controlled, operations approach. The "envelope" within which the spacecraft is flown can be gradually increased as the depth of experience from different seasons and mission environments grows. As detailed knowledge of spacecraft behavior and operational constraints grows, these can be incrementally incorporated into the planning validation process.

6) Automate all non-safety-critical processes. Incremental automation during routine flight operations can avoid wasted up-front investment and maximize the

return on effort according to the actual priorities seen during flight rather than those predicted during operations preparation.

7) Maintain readiness to swap between routine and exceptional operations. Critical mission operations cannot always be foreseen in advance, and some such as the MARSIS antenna deployments can end up being postponed well into the routine operations phase. Readiness to swap between routine and exceptional operations modes should be maintained well into the full design lifetime of the mission.

Acknowledgments

The authors would in particular like to acknowledge the contributions of M. Ladovaz and G. Mantovani to the development of the STAT software, and of M. Ricketts to the development of MIRA and S. B. Peschke for the development of the MPS++. These tools have significantly contributed to the science return from the Mars Express mission. We would also like thank the Project Science Team on Mars Express for their review of this chapter (F. Jansen, R. Pischel, T. Zegers, P. Martin, and D. Koschny).

References

[1] Porta, R., "MEX Science Data—From the Instrument to the PI," AIAA Paper 2006-5545, 2006.
[2] Rabenau, E., et al., "MEXAR – An AI Software Tool to Solve the Data-Dumping Problem on Mars Express," *SpaceOps 2006 Conference*, Rome 2006.
[3] Mounzer, Z., et al., "Flying Mars Express – A Day in the Cockpit," AIAA Paper 2006-56820, 2006.
[4] Ferretti, P., et al., "Mars Express Power Subsystem In-Flight Experience," AIAA Paper 2006-5857, 2006.
[5] McKay, M., Denis, M., and Hechler, M., "ESOC Operations Concept Workshop," ESA-ESOC Archive, Darmstadt, Germany, Feb. 2000 (unpublished).
[6] Denis, M., and Companys, V., "Mission Planning and Operations Workshop," ESA-ESOC Archive, Darmstadt, Germany, April 2002 (unpublished).

This page intentionally left blank

Chapter 31

Anomaly Recovery and the Mars Exploration Rovers

J. Matijevic* and E. Dewell†
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

I. Introduction

AS OF this writing (MERA sol 1034, MERB sol 1013, where 1 sol = 1 Martian day = 24.6 h), the Mars Exploration Rovers (MER) have each spent nearly three years on the surface of Mars. They each have driven over 6 km and encountered a variety of terrain conditions at the distinct landing sites of Gusev Crater (MERA, called Spirit) and Meridiani Planum (MERB, called Opportunity). They each have experienced a Mars year surviving and (sometimes) thriving in the change of seasons at the landing sites.

In the design of the rovers, variation in terrain, changing environment, and problems in operation from the surface of another planet were taken into account in the system. Although the implementation of the design is mainly "single string" (e.g., one computer system, one transceiver, one inertial measurement unit, etc., per rover), a degree of functional redundancy in the design makes it possible to operate successfully with one failure.

Uncertainties in the Mars surface environment and past experiences with landed missions led the project team to treat a MER mission as a temporary resource for scientific investigation. Early projections of a prime mission of three months and perhaps (with luck) an extended mission of three months thereafter led to an operations team concept that treated any day on Mars as a precious asset. As such the plan for each day would contain combinations of science observations and engineering activities [1].

A long-range planning team in MER operations prepares outlines of future activities taking into account assumed progress from data acquisition and changes

---

*Team Chief for Rover Engineering, Mars Exploration Rover Project. Member AIAA.

†Deputy Team Chief for Rover Engineering, Mars Exploration Rover Project.

to location of the vehicles. This "strategic" plan guides the operation through the creation of near-term objectives for vehicle activity. Each day a separate team (on each vehicle) prepares a set of sequences of commands that implement the objectives for a short period (1–3 days commonly) of the strategic plan. This tactical plan is a set of sequences prepared from "scratch" based on the best knowledge of the state of the vehicle, with only the constraints of times of communication and resources of data storage, time, and energy available. All sequences require development, assembly, and verification prior to uplink on the following day. After that uplink, the process begins again with the report of the results of execution; that is, a new vehicle status is established and reported to the engineers developing the plan for the following day.

When a day's activity reveals that a problem has occurred, the tactical planning leads meet to outline the strategy for resolving the problem and continuing, as possible. At times the anomaly may be as simple as a sequence that did not fully execute. Often when that sequence was intended to move the vehicle, the onboard protection may have determined that no safe path could be found to continue a move toward the goal. The response in the next day's tactical plan would be to replan the intended motion and continue toward the goal. If the anomaly shows that a component has indicated a failure to complete a required operation or has not shown expected performance, additional data may be requested or (in support of same) an engineering test scheduled. Typically other parts of the rover are unaffected by the component anomaly, and so the beginnings of corrective action (e.g., an engineering test) are simply added to other planned activities in the next plan. Sometimes little if any data were received from the execution of the last plan. This is in itself an anomaly, and a recovery response is planned in the next plan. This recovery plan emphasizes getting data and generally little else. Typically, when this recovery plan works, the next plan resembles one of the previously described responses. Lastly, when the anomaly analysis suggests a persistent component problem, an anomaly team is formed and (perhaps) a multi-sol investigation is initiated. This takes the corrective action out of the realm of a tactical response and often requires engaging experts to help in the diagnosis and recovery.

The following describes the problems that the two vehicles have experienced and the process used to resolve these problems. This description is preceded by a brief overview of the system design with particular emphasis on the flight software system that enables the problem detection and recovery. An overview of the ground operation for the MER rovers is also provided.

II. System Design

The MER is a solar-powered, six-wheeled-driven, four-wheeled-steered vehicle designed for operation on rugged terrain. Each vehicle carries an instrument complement consisting of a pair of narrow angled cameras [16-deg field of view (FOV)] each with independently actuated filter wheels (called Pancams), a miniature thermal emissions spectrometer (miniTES), and an instrument deployment device (IDD) to carry the four instruments: an alpha particle x-ray spectrometer (APXS), a Moessbauer spectrometer (MB), a narrow field (8 mrad) microscopic imager (MI), and a rock abrasion tool (RAT). The vehicle has six additional cameras: wide angled (120-deg FOV) base body mounted (called Hazcams, paired

front and rear on the vehicle) and intermediate angled (45-deg FOV) mast mounted (called Navcams). The mast holds both the Navcams and the Pancams while providing a periscope for the miniTES.

The instruments and vehicle equipment (motors, cameras, power bus, etc.) are wired to an electronics card cage called the rover equipment module (REM). The main computer is built around a RAD-6000 CPU (Rad6k), RAM and non-volatile memory. The non-volatile memory is implemented in a combination of Block erasable EEPROM (FLASH) and Electrically Programmable Read Only Memory (EEPROM).

Energy is collected from the solar array. This energy is channeled along the system power bus that is supported by two lithium-ion batteries. Power not required to support loads is channeled to recharge the batteries. The batteries support loads drawing power in excess of that supplied by the solar panel. Power in excess of loads and recharge required by the batteries is channeled to an external shunt radiator. The regulation and distribution of power is managed by the battery control boards (BCBs), one for each battery and independently powered by the batteries. The batteries also supply power to the mission clock with an alarm clock feature, programmable by software.

Temperature-sensitive components are located in the warm electronics box (WEB), an insulated compartment forming the body of a rover. Thermal control is primarily passive, with waste heat from electronics stored in the WEB during the day that radiates from the WEB during the night. Thermostatically controlled survival heaters and radioactive isotope heating units (RHUs) provide supplemental heating.

Communications are provided by two distinct systems: at X-band, a small deep space transponder (SDST), and two solid-state power amplifiers (SSPA), supported by a body-fixed, monopole low-gain antenna (LGA) and a high-gain antenna (HGA) steered in azimuth and elevation; and at ultrahigh frequency (UHF), a transceiver, supported by a body-fixed, monopole antenna. See Fig. 1.

The autonomous operation of the flight software [2] maintains the vehicle in the state needed to receive and act upon commands, execute sequences of commands when available, and collect and format data for transmission. Separate software modules handle certain engineering functions of power and thermal monitoring, power on/off of components, conduct of communications, management of the alarm clock, memory management, control of process execution, device health status, and performance of sequence control. Payload functions are managed in other software modules that acquire images, process science data, control actuators, power instruments, and coordinate multiple actuations as necessary to drive the vehicle or deploy the IDD. All payload functions are executed under sequence control with both timed actions and event-controlled operations available.

Because of energy limitations expected at times during the mission, the flight software was designed to support wakeups (i.e., boot of the CPU) and shutdowns as part of normal operations. A wakeup is scheduled once each day when energy production from the solar array can support the load associated with the CPU and supporting electronics. This wakeup is called the solar array wakeup that occurs at the production of about 2 A from the array that persists for at least 10 min. The BCBs determine that the energy production from the solar array meets the 2 A criteria and then can initiate a wakeup of the CPU and supporting electronics.

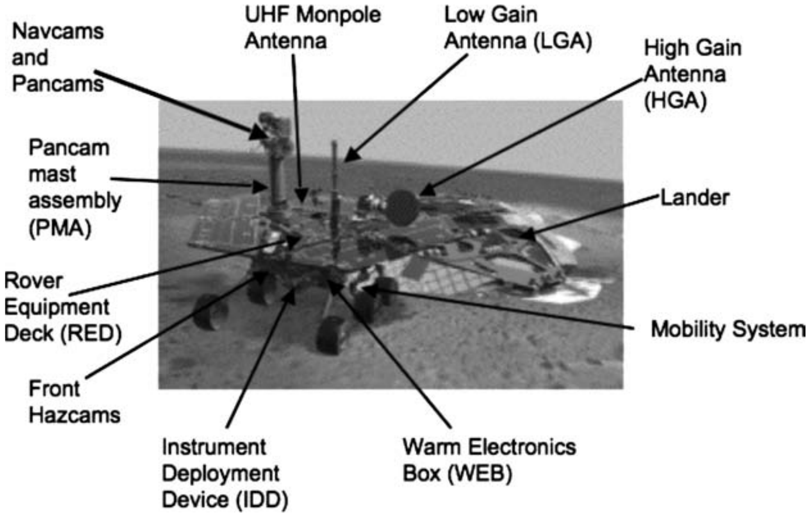


Fig. 1 Mars Exploration Rover.

A shutdown is controlled by parameters, established to ensure a power and thermal balance for the vehicle on any day when operations (including wakeups and shutdowns) are not otherwise commanded. A wakeup or shutdown may also be scheduled by sequence.

Communications are scheduled through the use of timed events maintained in an onboard communication windows table. At any given time this table contains about six weeks of timed events (windows) when uplink (commands transmitted to the vehicle) or downlink (telemetry transmitted by the vehicle) is planned to occur. The table contains both X-band and UHF communication events in a given period.

At each shutdown by flight software, a time is chosen for the next wakeup. This wakeup may be a sequenced wakeup, the beginning of a communication window, or an autonomous event scheduled by the flight software. Choosing the nearest (in time) event, the flight software writes the time to the alarm clock, enables the clock to perform a wakeup, and then performs a shutdown (opens the power-on switch). At the next wakeup time, the alarm clock, independently powered by the batteries, can issue a wakeup through the BCB. The mechanism the BCB uses to wake up the CPU and electronics is the same as the solar array wakeup: a power-on switch is held closed for a time period sufficient to allow the flight software to initialize and reinforce the power-on switch.

While in operation, the flight software records progress in execution through the generation of three types of telemetry: event reports (EVRs), engineering housekeeping and analysis (EHA) data, and data products. When exceptions in processing occur, often all three types of telemetry play a role in announcing the occurrence and documenting the circumstances leading to the exception. The flight software has several levels of response to exceptions. At the lowest levels, an EVR (warning) may be written in the record to note the occurrence. At the next

level a fault may be declared and the ongoing process or sequence ended. EVRs, EHA, and perhaps a fault data product are generated for the record. Other processing will typically continue. At the highest level of response, the flight software will autonomously reboot when an unrecoverable problem is encountered. This is a fatal exception. Because of the seriousness of the problem, there is not time to document conditions at the time of the fatal exception. The flight software addresses this problem by temporarily storing a small number of EVRs in EEPROM during execution so that these can be recovered after the reboot has been performed after a fatal exception. The reboot puts the flight software into autonomous operation and processing continues as described.

A fatal exception may also occur if a watchdog timer, used in monitoring progress in execution, is not refreshed. Once flight software has completed wakeup, there is one watchdog timer that must be refreshed by flight software three times in each interval of execution [real-time interrupt (RTI)]. Failure to perform this function leads to a reboot. Prior to wakeup, a watchdog timer tracked by the BCBs must be reset within 4 min of power-on. If the flight software does not perform this refresh function, the BCBs will power-off the CPU and electronics.

One additional autonomous feature of the flight software was added during MER surface operations. This feature, termed deep sleep, causes flight software to remove the batteries from the power bus, either autonomously or by command. This operation mode can be invoked once the sun no longer illuminates the solar arrays. In this mode only the mission clock is powered; all other loads on the power bus are removed. This mode of operation ends when the sun illuminates the solar arrays. A time placed in the alarm clock function of the mission clock that is later than the time of illumination of the solar arrays is honored, causing a wakeup of the flight software.

III. Anomalies

The main anomalies on the MER vehicles are described in some detail in the sections that follow. For convenience, these are grouped into flight software anomalies, hardware anomalies, environment induced anomalies, and an anomaly of uncertain origin. All anomalies were observed through the execution of the functions of the flight software but later attributed to purely a software fault, hardware failure, or an unaccounted change to the environment. The final category is comprised of the reset anomaly on Opportunity, an anomaly that has not been fully explained.

Conspicuous by it absence is any discussion of the FLASH anomaly [3] on the Spirit vehicle. This remains the worst problem experienced by either vehicle. The two weeks for recovery given a problem that impacted the normal operation of the flight software and thereby communications remain a testament to the capabilities and dedication of the MER operations team.

A. Software Anomalies

1. Race Condition, Initialization Counter

The first instance of this problem appeared on Spirit sol 131. At the time of the downlink of telemetry on the UHF-band, the vehicle was unexpectedly in

autonomous operation, that is, no sequences were active on board. Further a fatal exception was noted in the EVR log in the telemetry. In particular, the initialization module for flight software generated the EVR. After some investigation of the software code, the problem was seen to be a vulnerability that occurs when the initialization module was attempting to increment the counter of the number of times of initialization. This counter resided in non-volatile memory. Writing to this memory required permission from a separate software service that managed access to the memory. In the instance on sol 131, between the request and the grant of access to write to the memory location, another software module had requested, been granted access, and had written to non-volatile memory. The initialization module, finding it could not write the initialization counter, declared a fatal exception.

Within the structure of the flight software on the vehicles, all processes time-share the use of the single CPU. There is no guarantee that the three actions desired by the initialization module (i.e., request, being granted write access, and writing to memory) would occur contiguously. In this case, the vulnerability to a fatal exception would be viewed as a function of the number of processes in operation at the time of the write of the initialization counter. Clearly one corrective action could be to restrict the action of other software modules during this period of vulnerability. In part, for some payload modules, that action was implemented. However, the vulnerability was viewed as so limited in time (a few microseconds to perform the three actions desired by the initialization module albeit within about a 4-min window during initialization), the anomaly team reviewing the problem decided on issuing only an advisory for this vulnerability. The planning teams, noting the vulnerability of the motion of the IDD to an unexpected fatal exception, restricted use of the IDD from the 4-minute window during initialization. Otherwise, the likelihood of recurrence was deemed so slight as to not warrant further action, including consideration of any flight software change to attempt to correct the problem. This is understood as a race condition between software modules: a race that the initialization module won for many initializations (over 560 at that time) and many sols since this occurrence.

Recurrence proved not so unlikely as the problem occurred again on Spirit on sol 209 and then on Opportunity on sol 596 and sol 622. At the last recurrence, the fatal exception occurred during initialization in preparation for an afternoon UHF communication window. The flight software response caused the loss of that communication window. As a consequence, on sol 622 the downlink team received no telemetry, leaving the recovery team to sift through many possibilities for the problem. Eventually, at the next uplink window on sol 623, commands were issued and accepted by the flight software, and sequence control was reestablished. Sol 624 was a sol of normal operations.

Because of the delay in recovery of a sol after the sol 622 event, the anomaly team recommended enforcing a “keep out zone” for science operations after wakeup. Because of energy considerations, however, the recommendation was only enforced during the wakeup prior to an afternoon UHF pass. This ensured that a recurrence would not jeopardize the return of engineering and science data needed to plan for the next sol.

2. *Another Race Condition, Imaging Interface*

This anomaly occurred on Spirit on sol 136. There were no data returned on the sol 136 afternoon UHF pass. Plans for that sol and the next sol were such that there was a morning uplink communication window and afternoon UHF passes with two relay assets on sol 137: Mars Global Surveyor (MGS) and then Mars Odyssey Spacecraft (ODY). In an attempt to receive data at the earliest opportunity on sol 137, a command was issued to Spirit to make the uplink communication window a two-way uplink/downlink communication session on X-band. This action failed to result in a return of data.

When data were received from both relay passes, an analysis showed that the vehicle was unexpectedly in autonomous operation and a fatal exception was noted in the EVR logs. The fatal exception occurred when imaging sequences were deactivated prior to the preparation time for the afternoon UHF communication window [imaging cannot occur simultaneously with UHF communications due to electromagnetic interference (EMI)]. The imaging had not completed when the sequence deactivation commands were issued. The imaging software module, performing the image read/data write operations, was starting an image read when a command (part of the deactivation) was issued to power-off the hardware resource used in image acquisition. The software module, seeing the return of an error when attempting to access the hardware resource, declared an exception that led to the fatal exception response by the flight software. Since the fatal exception occurred during preparations for the afternoon UHF pass, the communications window could not be executed by the flight software (the window time was in the past when the reboot after the fatal exception occurred).

Further, when the reboot due to the fatal exception occurred on sol 136, the status of the system at the time of the fatal exception was not completely saved in memory. In this case, the position of the HGA antenna was not saved. This caused an X-band fault to be declared because the flight software had no knowledge of the position of the HGA when attempting to move the HGA in position for communications. This was the explanation for the failure to receive data during the attempt at X-band uplink/downlink communications on sol 137.

A modification to the deactivate sequence process was recommended after this incident. A directive to the imaging module to end execution was added prior to the power-off of the image acquisition hardware in the deactivation process. Once implemented, there has been no further occurrence of this problem. Normal operations were resumed on sol 138.

3. *Corrupted Command: Conjunction Test*

As a test of the degradation in command receipt experienced on the surface of Mars during solar conjunction (defined as the period at which the sun-Earth-Mars angle is less than 2 deg), commands were issued to the two vehicles each day during the period from MERA sols 244–255 and MERB sols 224–234. These commands were NO_OP commands, dummy commands issued to enable acknowledgment of receipt in the EVR logs generated by flight software. As the test progressed, more commands (at each command session tens of NO_OP

commands were issued in groups) were corrupted upon receipt as noted in the EVR logs. Finally, on MERB sol 229 a corrupt command caused a fatal exception. In analysis of the EVR logs and subsequent review of the command upload software module, a command with sufficient number of errors could be executed and cause a stack clearance to occur. This would explain the fatal exception.

Although the problem in the software could be corrected, the recommendation was to simply suspend commanding during the remainder of the conjunction period on both vehicles. Normal operations were resumed on sol 235 (first day after conjunction).

4. *Exception in Evaluation of a Defined Data Item During Mobility*

The design for sequence control supported by the flight software includes the definition of a defined data item (DDI) that can serve as a global flag for control of sequence execution. However, flight software clears this global flag after it has evaluated the flag in a conditional statement. There is only one such global flag, and so the definition and then evaluation should take place as a contiguous process. In the example of a set of drive sequences on sol 449, a DDI was defined in two sequences that were executed in parallel. The execution of this set of parallel sequences eventually resulted in an evaluation in one sequence following the definition in another.

The fatal exception that occurred was reported in the afternoon downlink in the UHF pass on sol 449. The drive that was implemented in the sequences of sol 449 ended prematurely, and none of the imaging typically performed after a drive was acquired [the reboot after a fatal exception places the vehicle in autonomous operation, and so no imaging sequences (as an example) commanded with the drive will be executed]. This fact plus the common practice at this time of the mission to avoid working weekends resulted in additional sols for recovery. Normal operations did not resume until sol 453.

A recommendation after this event was to restrict the use of a DDI to one sequence at a time. A flight software modification is planned to change the strategy of evaluation of a DDI, thereby removing the vulnerability to a recurrence.

5. *Upload Fault During Forward Link Commanding*

There had been no plan in the MER mission to use the forward command link capability of the UHF communication system: commands would be issued on X-band and telemetry would be returned on a combination of the X-band and UHF systems. The longevity of the MER missions and the exigency imposed by periods of absence of X-band coverage led the operations team to develop this capability while operating the two MER vehicles.

The first demonstrations of the facility for commanding a MER vehicle through ODY relay were conducted during the prime mission (within the first three months) and periodically within a year after landing. These demonstrations were implemented through the issuance of real-time commands that verified functionality of the command link. The first attempt at sending a full sequence load through the relay link was on MERA sol 603. This was a single sol plan with 15 sequences. All were received successfully. The second attempt on MERA sol 605

was a 3-sol plan with 41 sequences. Again all were received successfully, although in the EVR log there were a number of warning messages associated with processor loading. As understood at the time, these warning messages were a benign consequence of the CPU failing to "keep up" with the combination of forward link commanding conducted simultaneously with downlink through the UHF transceiver.

These demonstrations helped to prepare the operations team for a formal readiness test conducted in-flight on MERA with commands issued through the forward link system and executed onboard. The first forward link upload was a 3-sol plan containing 53 sequences commanded on sol 640. These were received successfully but again with warning messages as seen on the prior sequence loads. The second forward link upload was a 2-sol plan with 43 sequences commanded on sol 644. After receiving 12 of the 43 sequences, there was a fatal exception in the flight software. On analysis of the EVR logs and a review of a simulation of the events on the upload of sol 644 in a test bed for the MER vehicles, a combination of the number of sequences and a dropout in the forward link due to geometry between ODY and the vehicle resulted in the fatal exception. The remainder of the readiness test was cancelled and recovery to normal operations was achieved on sol 646.

This incident was followed by an effort to develop a workaround, robust to the number of sequence and the possible geometry-induced variability in communication link performance between the ODY relay and the surface vehicle. After testing with a MER test bed, a strategy for forward link commanding was developed: more "padding" (null characters) was introduced between sequences in the command files transmitted by ODY, and the number of sequences transmitted was reduced to not more than 25 with allowance for a few additional real-time commands. The increased padding in the command files reduced the processor contention during command decoding, verification, and storage while telemetry was being processed and transmitted. The reduced number of sequences was consistent with the protocol overhead and the command rate possible with the forward link. This strategy was demonstrated successfully by test sequence loads issued on sols 747 and 755. In both instances, 25 sequences plus an additional real-time command were issued. Forward link commanding was successful in practice when the uplinks on sols 773, 775, and 777 during a period of no X-band coverage for Spirit resulted in operations on sols 774–779.

B. Hardware Anomalies

1. *Stuck-On Heater*

Upon receipt of Opportunity's first overnight UHF pass on sol 2 at 03:30 LST (Local Solar Time), the power team reported that the nighttime loads from sol 1 22:54 LST to sol 2 3:30 LST were ~0.5 A larger than predicted. The subsequent pass showed that the additional load had remained on until ~10 LST, dissipating ~176 W·h. An anomaly team convened to develop a possible explanation and recommend a resolution. The highest likelihood fault was that a rover power distribution unit (RPDU) load was unexpectedly powered on. Because of the size and the on/off times for the load, the IDD heater 1 was identified as the source of the problem. The on/off times for the load corresponded to the predicted thermostat box switch times and to the temperature rise recorded by the temperature sensor

on the MI, located close to IDD heater 1 when the IDD is in a stowed configuration. Because of the design of the thermostat box, the IDD heater was not powered on during the day (~10 to ~23:00 LST). This prevented overheating in the event that an IDD heater circuit was stuck on. This design also suggested a mitigation strategy for the energy loss due to the stuck-on heater circuit; remove the heater from the circuit at night.

The rover was still completely functional as designed. However, the energy drain from the heater would reduce the energy available for science activities. As winter approached, when less energy was available, the activities on the spacecraft would be much more limited, and the survival of the spacecraft would be at issue. The solution (removing the heater from the circuit at night) was implemented by arranging to remove the batteries from the power bus. By removing the batteries from the power bus, the BCB was also not powered, leaving only the mission and alarm clocks powered. At dawn, the BCBs are powered (awakened) simply by having sufficient light impinge on the solar arrays (about 0.2 A of current generated by the array). The net savings would be ~180 W·h/sol. The team implemented this solution, termed "deep sleep," for the first time on sol 101–102. Deep sleep was permanently enabled on sol 206, so that it was the default state unless temporarily disabled for the night. One drawback was that survival heaters for the miniTES and REM, which are normally left on during the night, were taken offline. Deep sleep is currently used on average every other night on Opportunity.

The use of deep sleep to mitigate the stuck-on heater energy drain enabled an extended mission for Opportunity. It came at a price, however. With batteries removed from the power bus, no survival heater for the miniTES could be powered over a night of deep sleep. The colder temperatures on the miniTES (routinely below the acceptable flight temperature limits) due to deep sleep have undoubtedly contributed to a degradation of this instrument on Opportunity (see the section on resets in the following). This degradation has not been experienced by the miniTES on Spirit. Because of the generally colder temperatures at the Gusev landing site and the absence of a significant energy drain, Spirit has never used deep sleep.

2. Right Front Drive Actuator

Spirit's drive from Bonneville crater to the Columbia Hills was planned to satisfy the twofold objective of reaching potentially a new geological target for investigation and enabling the vehicle to spend the winter at northerly tilts (of benefit for vehicle safety for a solar powered system). The drive was nearly 3 km and needed to be completed in a period from sols 86–156. The drive was accomplished by driving on 50 of the possible 70 sols. This was a remarkable accomplishment for Spirit at that time in the mission, but it was achieved at the cost of increased current draw seen on the right front drive actuator. The drive actuator for the MER rovers was designed as a geared, lightly lubricated system. Each drive actuator required on average approximately 0.4 A during motion of the vehicle. The current draw can spike when the vehicle is engaging an obstacle, but these spikes are generally less than 1 A for a fraction of a second. By the end of the period of the drive to Columbia Hills, the right front drive actuator required nearly

1 A while the vehicle was in motion with spikes over 1.2 A. Further, the current draw for this actuator had increased exponentially during the last 10 drives. At that rate, additional degradation would lead to a drive actuator that could not be supplied sufficient energy to move. It was estimated that this loss of the drive actuator could occur within the next 100 m of travel.

An anomaly team was formed to review the data and the possible options for recovery. The team proposed stopping the drive at the base of the hills and performing a series of motor diagnostics. A relatively flat area was chosen for these diagnostics that consisted of small forward and backward motions with the actuators warmed prior to the operation. The only slight improvement in performance of the right front actuator after these tests resulted in the team developing a strategy for conserving further use of the actuator. The strategy involved driving the vehicle "backward" while dragging the right front wheel. Periodically, the right front wheel would be used to help correct the error in direction created by having the vehicle move while dragging the right front wheel. With this strategy, Spirit climbed into the hills and continued its mission. Over the period of the succeeding 200 sols, the combination of reduced use of the right front drive actuator, driving backwards, warming the drive system before driving, and simply standing still, usually at a spot where in situ investigations were conducted, resulted in the right front drive actuator eventually returning to operation (i.e., current draw) comparable to that seen on the other drive actuators. The problem with the actuator was attributed to the flow of the lubricant in the gearbox. The persistent driving in one direction "starved" the gearbox of lubricant, causing the higher, anomalous current draw. Warming the actuator, reversing the direction of driving, and not driving every few days was sufficient to correct the problem.

In this period of about 200 sols after the problem was detected, only 4 sols were devoted to engineering tests designed to diagnose the problem and practice the strategy of driving while dragging a wheel. Science observations were conducted and the vehicle was driven (albeit for short distances) in a nominal fashion while the anomaly team was performing analyses and developing the strategies to conserve right front actuator usage.

3. *Right Front Steering Actuator*

On Opportunity on sol 433 in the middle of a drive, the right front steering actuator stalled. In the period from sol 358 to sol 433, Opportunity had traveled nearly 3 km, driving on about half of the days in that period. The stall occurred without warning, that is, the steering actuator was used nominally on the prior segment of the drive. On the drive segment in question, the steering actuators were being positioned for a turn when the steering actuator current peaked to the preset safety limit (2 A).

An anomaly team was quickly formed to review the images acquired during that day and the data logs at the time of the incident. A test was proposed and executed on the next sol. The test involved commanding the steering actuator clockwise then counterclockwise a few degrees at several voltage settings to test for any motion. There was little movement and that motion only at the highest voltage setting during the test. The anomaly team concluded that there was an obstruction in the actuator, likely at the first gear stage. As such, the motor alone

has little torque to move the obstruction. The team recommended leaving the actuator at its current position and operating nominally. Fortunately, when the motion of the actuator was stalled, the right front wheel had been steered only about 7 deg from the nominal straight-ahead position. In subsequent drive operations, the planning team pivots about the right front wheel using larger arc turns rather than turns-in-place to position the vehicle.

Normal operations of the payload resumed on sol 434, and driving continued within a few days (sol 437). At this writing, the strategy of pivot turns around the right front wheel and variants such as “K” (or three point) turns continues to be used.

4. IDD Azimuth Actuator

On Opportunity on sol 654, the IDD was commanded to deploy from its location below the WEB shelf. The first motions of the IDD involve the movement of the elevation and elbow actuators that uncouple a hook on the elbow assembly from a support roller. After this operation is completed, the shoulder azimuth actuator moves the hook away from the roller in preparation for a subsequent movement of the elevation and elbow actuators to move the IDD away from all support structure below the WEB shelf. The shoulder azimuth actuator did not move in this sequence, causing the operation to fail.

An anomaly team was formed to consider tests that could be conducted to both move the shoulder azimuth actuator and collect data during that motion. Changes in control parameters that define the movement profile of that actuator were proposed and planned for execution on sol 659. There was little motion on that attempt. Changes to these same control parameters, effectively allowing for more current to be supplied for longer periods during the motion, were attempted on sols 660 and 661. There was little motion on either attempt but what motion was seen suggested that the motor resistance had increased significantly from the nominal value calculated during prior (and successful) unstow operations. A motion with a control parameter of a higher resistance was planned for execution on sol 666. This motion succeeded, and the detailed data verified that the motor resistance had doubled. With the increased resistance value as a control parameter, the IDD was deployed successfully on sol 671. Subsequently, a science investigation using the IDD and planned at the location of Opportunity on sol 654 were carried out. The motion of the IDD and, in particular, the performance of the shoulder azimuth actuator were somewhat restricted and a little unpredictable. However, the experience in operation at this site gave the operations team a performance characterization that was used at other sites in the vicinity. This experience also provided the foundation for the operation of the IDD from this point forward in the mission.

A past characterization test conducted with the type of motor used in the implementation of the shoulder azimuth actuator revealed a failure mode in which the motor resistance had doubled due to an open wire in one of the windings. There are two windings in this brushed motor. A second open wire would cause the motor to fail. A likely cause of degradation in the motor winding is thermal cycling. This actuator is on the stuck-on heating circuit (see Sec. III.B.1), and the temperature transient is nearly twice the scale of that of all but one other actuator

on the IDD and elsewhere on the vehicle as a whole. The other actuator is the elevation actuator of the IDD that shares the problem of the stuck-on heating circuit.

The anomaly team discussed how the vehicle could drive with the IDD with a degraded shoulder actuator. If the IDD was stowed under the WEB shelf and the shoulder azimuth actuator failed, the IDD and all instruments for in situ measurement could no longer be used in the mission. If the IDD was deployed, how much driving could be allowed and what terrain could be traversed so that the vehicle motion did not damage the IDD or (at a minimum) cause a loss in calibration for the IDD? A position, with the wrist and turret suspended over the solar array (the "hover stow" position), was analyzed and shown to tolerate drive excursions that involve differential motions of the vehicle of about 4 cm. Several targets were accessed by Opportunity driving with the IDD in the hover stow position. For longer drives, an evaluation of the terrain to be traversed cannot ensure that the differential motion constraint is satisfied (e.g., not all areas planned to be traversed in a long drive can be imaged). Instead, the anomaly team agreed that the IDD must be stowed in the usual position below the WEB shelf for any long drive. Because the open wire anomaly in the winding of the shoulder azimuth joint is understood to result from thermal cycles, it was recommended that the IDD be stowed prior to the drive then deployed immediately after the drive. The deepest thermal cycle would then be avoided with the IDD stowed below the WEB shelf. The first successful drive using the stow/drive/unstow strategy occurred on sol 731. At this writing, there is additional analysis planned for the eventuality when the IDD can no longer be stowed below the WEB shelf (i.e., another open in the winding for the shoulder actuator occurs, causing loss of that actuator; degradation or loss of the elevation actuator). Driving may be problematic at best at that time.

Throughout the period from sol 654 to sol 731, each sol contained nominal operation of other payload elements, and after sol 671 the IDD was once again scheduled in science operations. See Fig. 2.

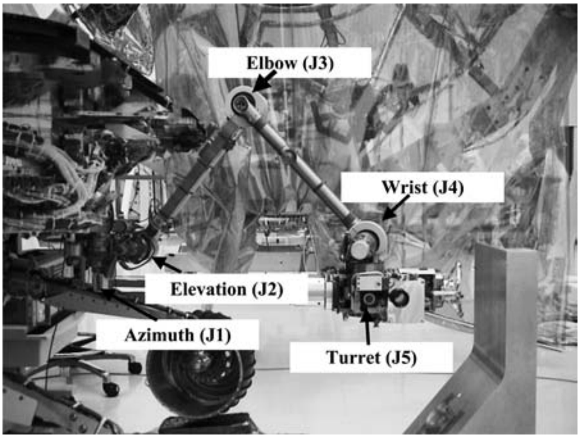


Fig. 2 Instrument deployment device (IDD).

C. Environmentally Induced Anomalies

1. Clock Fault

On the morning of sol 628, Opportunity encountered a dust storm that significantly increased the tau value (optical opacity of the sky) from 0.82 to 1.8. This increased "haziness" in the sky caused the sunlight to be more diffuse, and thus provided less power to the solar arrays. The result was that both the BCB wakeup and solar array wakeup occurred later than the uplink team expected. The plan on the evening of sol 627 was to invoke deep sleep, and schedule a sequenced wakeup at 07:40 to re-enable the miniTES heaters. The previous 2 sols' BCB wakeups were at 07:23. However, due to the dust storm, the BCB wakeup on sol 628 occurred just after 07:40. Thus, the 07:40 alarm clock occurred while the batteries were still offline, and nothing responded to the wakeup request. The next time the rover had a wakeup was at the solar array wakeup at 10:44 LST, at which time the alarm clock time was in the past, and so the flight software deactivated all sequences and thereafter was in autonomous operation. The next time given to the alarm clock was for the start of the afternoon UHF pass, which was the time at which the operations team learned a problem existed. After the UHF pass, the alarm clock set the rover to wakeup at 18:30 LST to do a "wakeup for deep sleep" followed by a shutdown bringing the batteries offline (i.e., deep sleep is the default in autonomous operation). However, the onboard communication table had a morning pass, as the plan for the evening of sol 628 and morning of sol 629 was to disable deep sleep for a single night activity. This resulted in the alarm clock being set for that morning pass and, as a consequence, a clock fault occurred again that following morning of sol 629. This fault did not affect the flight software that was already in autonomous operation from the fault on sol 628. The only impact from this second fault was that due to the residual effects of the dust storm a late solar array wakeup at 11:18 LST caused the loss of an X-band communication window that was supposed to occur from 10:30–11:00 LST. A somewhat mystified operations team later worked out why no data were received from that scheduled X-band window. Because this was part of a 3-sol weekend plan, and the vehicle was otherwise in a power positive (and eventually) understood state, the operations team waited until the following Monday plan (sol 630) to resume nominal operations.

The uplink team decided to not wake up after deep sleep to enable the miniTES heaters for the next few sols until the dust storm had dissipated to prevent missing the BCB wakeup time after deep sleep. Once the skies cleared, the planning team established a guideline to schedule wakeups 45 min to 1 h after the BCBs come online.

2. "Potato" Rock

On sol 339, Spirit's rover drive planners had planned a drive up Husband Hill in route to an area deemed scientifically interesting and an ideal place to take panoramas of the hill traversed thus far. The progress had been slow, with the rover encountering high slip and surrounded by a rocky hillside. At the first turn command in the drive sequence, the rover was commanded to turn in place 15 deg in 17.5 s. In the last 2 s of the turn, the right rear wheel current spiked to 1.8 A

Fig. 3 Potato rock stuck in left rear wheel.

(nominally the wheel actuator draws about 0.35 A) and the drive motor actuator stalled. A rock was lodged between the inner ring of the wheel and the actuators, as could be seen in the imagery transmitted on the sol 339 afternoon UHF pass (see Fig. 3). On the following sol, the operations team dislodged the “potato” rock (so called due to its size and shape) by spinning the right rear wheel in the opposite direction of the sol 339 drive. The rock was dislodged from the actuator, but remained inside the wheel. The next challenge was to get the rock out of the wheel, and the operations team decided to try dragging the wheel in an arc. On sol 343, the right rear wheel was straightened, and then the drive and steering actuators were disabled (a temporary conditional setting in flight software). Two small 0.3-m arcing drives were attempted using the remaining five wheels. However, the rock remained in the wheel. On sol 344, the rover was commanded to back down the hill (all actuators enabled this time), but the rock remained in the wheel. On sol 345, the rover was commanded to turn in place, to drive back down the hill, and to perform a final turn in place. The sequence of motions completed successfully, but the potato rock was still in the wheel. Finally, on sol 346, two drives and one more turn in place were commanded. On the afternoon UHF pass, images confirmed that the potato rock was out of the wheel. Nominal operations resumed for the weekend plan on sols 348–350.

During development of the rover mobility system, the design considered the potential of small rocks kicked up by vehicle motion into portions of the drive mechanism. Rock jamming in the mobility mechanisms had occurred on prior rover systems with the result that in the implemented MER system there was additional external clearances around the wheels and the drive actuator mechanisms

were located within the wheel well. The wheel wells were not "closed out" with structure due to weight considerations for the vehicle as a whole. Despite these provisions, the geometry of the vehicle and the loose rocks and regolith of the terrain on Husband Hill made this rock jamming possible.

3. *Embedding in Terrain*

On sol 446 Opportunity drove into a dune. The drive was planned for 90 m with almost all of the drive executed with the vehicle driving backwards. This drive was the 36th segment of a drive across the Meridiani plain that had already covered over 3 km in distance south from the vicinity of the landing site. As were many of the prior drives, the drive on sol 446 was conducted "blind" with few safety checks employed during execution. The environment had enabled this type of driving since a generally flat, featureless terrain was presented to the operations team on each of the past drives, including the terrain imaged on sol 439 and used in planning for the sol 446 drive. Also, ripples in the regolith and small dune features seen in previous terrain images had posed no hazard for the mobility system for Opportunity. The ripples and small dunes were unremarkable in the images of the terrain used for planning the sol 446 drive, and these features were not considered any hazard for the vehicle. About 40 m of the drive on sol 446 had completed when the embedding in the dune began. The vehicle was slipping and embedding itself in the dune while completing the final 50 m of the drive. A final turn at the end of the planned drive to achieve the best position for return of data in a communication session was not completed, and only at that time was a fault declared by the flight software. The material transition was over 2 m, and the rover climbed upon a dune over 30 cm in height. The final images returned on the UHF pass on sol 446 showed the wheels about 70% buried (see Fig. 4).

Upon review of the data from this drive, the operations team was mainly concerned that the last segments of the drive and the turn had resulted in little change in the rover position. Also, on inspection, the wheel cleats were "caked"

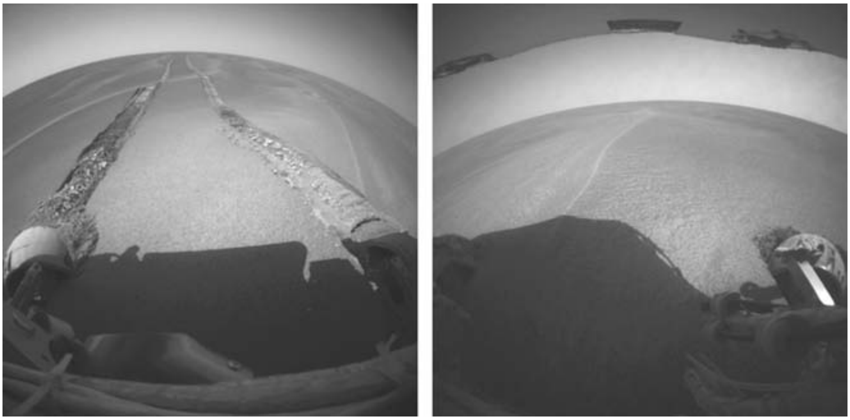


Fig. 4 Front and rear views of opportunity embedded in a dune.

with regolith. During development of the MER mobility system, tests showed that the cleats played a significant role in the engagement of terrain, modifying the ground and creating mechanical "purchase" that allowed forward motion of the vehicle to occur. Wheels without cleats experienced slip in otherwise benign, sandy terrains. Would Opportunity be able to escape this dune without exposed cleats and regolith piled to the top of the wheels?

An anomaly team was formed with individuals who had participated in the test and eventual qualification of the MER mobility system. Added to this team were science team advisers with understanding and test experience with vehicles driving in Earth-based terrains. Tests with test bed vehicles driving out from beach sand were promising: the test bed vehicle drove easily out of the terrain beginning with wheels buried completely in the sand. From soil tests conducted during the earlier phases of the mission, the team surmised that the Meridiani regolith was likely a less cohesive mixture of material than beach sand. A variety of materials were mixed with the objective of making something that behaved like Meridiani regolith. Such a mixture of materials could be used as a simulated regolith in test with the test bed vehicle. Eventually, a mixture of diatomaceous earth, mason clay, and sand proved to have a consistency like that seen in soil tests on Mars. This material had the added benefits of being generally available and certified for personnel exposure in test facilities. An area in the MER test facility was prepared with this mixture, and the test bed vehicle was driven into it. A series of tests were conducted with the wheels buried at the levels seen on Mars with Opportunity. In each case the test bed rover drove out of the material, although with difficulty. About 50 m of wheel motions were required to move the test bed vehicle about 2 m out of the material with most of the motion occurring at the end of the drive.

With these tests completed, the operations team began to drive Opportunity out of the dune. Driving the vehicle out of the dune would initially be in short movements, designed to determine if there was any chance that vehicle motion would cause further embedding into the dune. The first moves (a wheel straightening and a few meters of driving) began on sols 461 and 463. Subsequent drives through sol 468 resulted in less than 10 cm of forward motion per each drive of about 8 m or roughly 99% slip. After sol 469, 8 m then 12 m then 20 m of motion was commanded until the vehicle was extract from the dune. In total, 2 m of forward travel required almost 200 m of commanded motion with the last meter of travel occurring on the final sol.

In this period, the operations team developed then tested strategies that provide protection from embedding events on future drives. These strategies included measured current draw vs forward motion, use of visual registration of movement to determine slip in forward motion (applied on any drive greater than 5 m), incremental turns to avoid large displacements of regolith by the wheels, analysis of terrain to predict successful traverse, and drive length reduced to 30 m or less (i.e., driving within the imaging range of the vehicle). The dune of this embedding was 35 cm tall: the largest dune formation seen in the terrain prior to sol 446. Imagery from this location and thereafter showed that dunes of this size were increasingly common as the vehicle moved south. Reconnaissance imaging would be a part of any drive planning from this point forward.

While driving was limited during this period from sol 446 to sol 484, science observation continued throughout this time. The mobility analysts and rover drive

planners on the operations team developed and carried out the testing program in the test bed. They developed then implemented the drive strategies that have resulted in about 1.5 km of successful driving since that time.

D. Anomaly of Unknown Origin

On each of three sols (sols 440, 563, and 610), Opportunity experienced an unusual reset event: an unplanned shutdown then wakeup. Each of these events occurred during the execution of a miniTES observation, and little data were reported at the time of wakeup from each event. In the case of the reset event on sol 440, the system response was consistent with a fatal exception in flight software, that is, an autonomous reboot occurred. However, in this event, there was no EVR reported for the fatal exception, and those EVRs stored prior to the exception (as a short history of execution) shed no light on the cause. The cause of the reset was reported in an EVR received on the subsequent wakeup as the watchdog timer failing to be refreshed. This is the watchdog timer used as a self-test while flight software is executing. In the cases of the sol 563 and sol 610 reset events, the system response was consistent with a watchdog timer prior to wakeup not being refreshed and a reset was initiated by the BCB. In both cases, a miniTES observation was initiated (a sequence was activated) prior to completion of the initialization process by the flight software. Such sequence activation is allowed by the flight software and has been successfully executed many times during the mission of both rovers.

At the time of the reset event on sol 440, an anomaly team was formed to consider the possible causes. Without data related to the cause of the event (why was the self-test watchdog timer not refreshed?), the focus of the investigation was the consideration of the possible collateral effects associated with the continued use of the miniTES. Through test on the vehicle, the Pancam mast assembly (PMA) and associated motor controller were determined to not be causes of the problem (other payload items successfully used these components and the associated electronic interfaces subsequent to the sol 440 reset). The electrical interface to the miniTES instrument was reviewed for possible failure effects. This review showed that no short or open in the wiring to the instrument would cause a severe cascade, damaging other equipment on the electronic board providing the interface to the miniTES instrument or other portions of the REM. Other possible causes principally associated with flight software in operation (i.e., a VersaModule Eurocard (VME) bus error, a spurious uplink command, a software controlled reset, an under-voltage in a power converter unit, and a BCB watchdog timer induced reset) were all reviewed and discounted due to a lack of evidence of occurrence. A self-test watchdog timer failure could be the result of a software module "hanging" in execution. However, the software modules in execution at the time were routine health and maintenance tasks of flight software, motor control, and the miniTES payload interface module. All these modules had performed many similar operations prior to this point (and thence thereafter) without incident. A code review of the miniTES payload interface module revealed no particular vulnerability for hanging in execution.

After a recovery sol where component states affected by the reset event and interfaces to the PMA were tested and shown to be working nominally, normal operations (without observations by the miniTES) were resumed on sol 442. After the review of both hardware and software was completed, the anomaly team

concluded the miniTES was not a significant risk either to the instrument itself or to other rover equipment. On sol 487 routine miniTES observations were again scheduled on Opportunity.

At the time of the sol 563 reset event, the anomaly team considered the reset more serious than the sol 440 reset event. The BCB initiated reset could be associated with a problem in the rover power system or in the BCB itself. Further, since a BCB initiated reset does not automatically reboot the flight software system, the next software wakeup could be an alarm clock timer wakeup scheduled 26 h in the future. The wakeup process loads 26 h from the time of last wakeup into the alarm clock as a precautionary value intended to allow the rover system to recharge batteries in autonomous operation prior to attempting to wakeup and initiate flight software. This design covers a variety of possible power system anomalies, all of which could result in a flight software failure to wake up successfully as indicted by a BCB initiated reset. If this value in the alarm clock is not changed by a subsequent successful flight software wakeup, the rover system could spend a sol or more not communicative with the MER operations team and not under sequence control.

Fortunately, in the sol 563 reset event (as was also the case in the similar sol 610 incident), the automatic solar array wakeup was the next wakeup event. The flight software in autonomous operation after a successful wakeup honored subsequent communication windows, including the afternoon UHF window on sol 563. At the time data were reported from sol 563, the system had returned to predictable operation. On the succeeding sols a slow recovery to nominal operation was planned. Execution of sequences that exercised the PMA, the mobility system, and various health and maintenance functions of the flight software were performed. No problems were observed with any of these functions and nominal operations (without the miniTES) resumed on sol 571.

The anomaly team had few leads for determining a source of the problem. There were no data recorded during the event or reported from the time of the event. The timing was such that the miniTES observation was only initiated before the BCB induced reset occurred. The team reviewed the software implementation of the BCB but could determine no particular problem with the watchdog timer function. The rover system performed as designed, but the question remained: what caused the reset? As recommendations, the anomaly team requested that if the miniTES was used again, its sequences should be executed outside of the period of wakeup initialization of the flight software. This operation restriction may cause a subsequent reset to occur in the manner of the reset event on sol 440. Also, the team recommended that an attempt be made to replicate the circumstances of the system response to the reset on one of the MER test bed systems. There was no miniTES instrument built for use in a ground test facility. A simulator was provided during MER software development that included an interface to the RS422 (serially balanced and differential bus interface standard) standard used on the electronic board that provided a power and data connection to the miniTES instrument in the flight configuration. The miniTES payload interface module had been developed using this simulator, and there was no record of a problem such as this software hanging during execution: a possible reason for a watchdog timer to not be refreshed. Perhaps applying an external stimulus at this interface to the simulator in the test bed could cause a reset response like that on Opportunity.

In lieu of the results from any test bed investigation, the miniTES was still considered no significant risk either to the instrument itself or to other rover equipment. On sol 570 a miniTES observation was again scheduled on Opportunity. With restrictions (e.g., scheduling miniTES sequence execution after completion of wakeup initialization of flight software), miniTES observations were regularly scheduled after sol 577. The operations team assembled a standard recovery plan on the chance that a reset could occur in the future. This plan reduced the time of recovery to a single recovery sol that contained the tests necessary to determine that nominal operations could resume. Because of successful execution of the miniTES from sols 577–598, all miniTES restrictions were lifted thereafter.

The decision to allow unrestricted miniTES usage proved to be wrong with the recurrence of the reset event on sol 610. As was the case on sol 563, the BCB reset the flight software prior to wakeup due to a watchdog timer failing to be refreshed. A miniTES sequence was just activated when the reset occurred. Because of the similarity with the reset event on sol 563, there was no reconstituting of the anomaly team. In contrast to the response to the reset event on sol 563, the MER operation team applied the streamlined recovery plan on sol 611, and Opportunity was ready to resume nominal operations on sol 612. No further miniTES observations were scheduled until sol 656. At that time only one observation was allowed per week based on an agreement with the principal investigator and the project manager for MER operations. This restriction was lifted on sol 728 when the test bed investigation was completed.

While the operations team was conducting the recovery leading to normal operations on Opportunity, an engineer, consulting with members of the anomaly team, prepared one of the test beds with break-out-box and signal generator at the interface between the miniTES electronic simulator and the REM data and power system. Tests of interruption in the power and data signals to the electronic simulator were performed while the flight software was in operation consistent with the use of the miniTES during the reset events. After many trials, a test case resulted in the triggering of a hanging of the flight software due to the interruption of signal while the flight software was in the midst of the initialization process. The hanging caused a BCB initiated reset, similar to the sol 563 and sol 610 events. At other times (generally after most of the initialization had been completed), the interruption of signal resulted in the declaration of a fatal exception. This fatal exception was accompanied by a warning EVR that pointed to a portion of the processing in the miniTES payload interface module. This processing used the zero path difference [(ZPD), a mean estimate of the size of the spectra generated during an observation by the miniTES] as a parameter in the calculation of compression of the data produced by a miniTES measurement. In the test case the ZPD value was zero, causing a software cascade leading to the hang. Repeating the test with versions of this software module with a fixed (though artificial) nonzero ZPD value or without execution of the portion of the processing involving the ZPD in data product production resulted in no occurrence of the software hang and the subsequent BCB initiated reset or fatal exception.

So why should zero ZPD values ever occur? Why should this occur on Opportunity and not Spirit? Why should this occur at this late stage of the mission? The explanation for this may be traced to an incident and subsequent corrective action that occurred only on Opportunity and only after the miniTES experienced

a pronounced degradation. At that time (after sol 394 on Opportunity), the instrument routinely produced short interferograms (a representation of the spectral output of the instrument) that the flight software rejected for production into data products and transmission to Earth. The analysis of the problem at that time (from a consensus of a combined contractor, university, and MER engineering team) was that the instrument had suffered degradation in the servomechanism used in control of the scan mirror within the instrument. The cause was likely the result of repeated cold cycles associated with the use of the operation of deep sleep. This technique of deep sleep, required to continue the mission of Opportunity beyond the prime mission period, exposed the miniTES to thermal cycle below the allowable flight temperature of –40ºC. The resolution at the time of the sol 394 and subsequent incidents was a software change that accepted short interferograms in the production of miniTES data products. Such short interferograms can have zero ZPD value in an anomalous case.

Unfortunately, the data collected from the reset incidents on Opportunity do not include a data product from the miniTES at the time of any reset event. Associating the test case with the flight incidents may be an explanation for the resets but cannot be viewed as root cause for the incidents. The test case did reveal a vulnerability that was corrected with an operational restriction on the type of data product produced by a miniTES observation. This restriction was imposed after sol 728. In addition, the restriction that miniTES sequences should be executed outside of the period of wakeup initialization of the flight software was enforced after sol 728. There have been no subsequent incidents in the subsequent year of operation.

IV. Conclusion

The MER operation process was developed to allow the rovers to drive every sol. Since at least the terrain environment changes when the vehicle moves, the results of a drive require evaluation before the rover can be safely commanded to drive again. This requirement led to scheduling command interactions and telemetry return every sol during the mission. This process (daily evaluation and planning on the tactical timeline) proved to be beneficial in also training the operations team to respond on a tactical shift to the uncertainties in command execution and to changes to the assumptions about continued rover operation. This is the training necessary for recognizing and responding quickly to anomalies.

The major anomalies experienced by the rovers are described, and the responses of both the tactical operations team and, when necessary, an anomaly team are detailed. In general, the operations team continued payload operations within a few days after every one of these events. The anomaly team, cognizant of the need to continue the science observations of the missions, proposed engineering tests and changes in vehicle procedures that fit with nominal science operations.

The missions for Spirit and Opportunity continue. Since the time of the reported anomalies above (about sol 730 for each vehicle), a new software image has been loaded. The software corrects problems with the recording of HGA positions described in Sec. III.A.2, the management of DDIs described in Sec. III.A.4, and corrects the logic using the ZPD value as described in Sec. III.D. There have also been additional anomalies on both vehicles. The process that was used to address

the problems described has been employed in these additional cases with little interruption of the science operations.

Acknowledgments

The work described in this chapter was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

References

[1] Mishkin, A., Limonadi, D., Laubach, S., and Bass, D., "Working the Martian Night Shift, The MER Surface Operation Process," *IEEE Robotics & Automation Magazine*, Vol. 13, No. 2, June 2006, pp. 46–53.

[2] Reeves G., and Snyder J., "An Overview of the Mars Exploration Rovers' Flight Software," *Proceedings of the 2005 IEEE Conference on Systems, Man, and Cybernetics*, Waikoloa, HI, Oct. 10–12, 2005, Vol. 1, pp. 1–7; also http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=33254&is Year=2005.

[3] Reeves, G., and Neilson, T., "The Mars Rover Spirit FLASH Anomaly," Institute of Electrical and Electronics Engineers, Paper 1425, 2005.

[4] Neilson, T., "Mars Exploration Rovers Surface Fault Protection," *Proceedings of the 2005 IEEE Conference on Systems, Man, and Cybernetics*, Waikoloa, HI, Oct. 10–12, 2005, Vol. 1, pp. 14–19; also http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=33254&is Year=2005.

Chapter 32

Preparing Mission Operators for Lunar and Mars Exploration

Stephen M. Parkes,* Martin Dunstan,† and Iain Martin‡

Space Technology Centre, University of Dundee, Dundee, Scotland, United Kingdom

and

Salvatore Mancuso§

European Space Agency, Noordwijk, The Netherlands

I. Introduction

FUTURE European planetary missions will include rovers to explore the surface of Mars and other planetary bodies. Mission operators will need to be trained to operate the rovers and to cope with various problems that could possibly occur during the mission. Operators can be trained using a prototype of the rover and a physical mockup of a Martian surface, along with simulation of the communication delay. A significant problem here is the availability of the prototype rover for operator training. Also the training facility would be large and necessarily limited in scope. Another problem is that providing support for training for fault recovery can be challenging on a real system as introducing the fault can be problematic. Another complementary method of operator training is to use a simulation of the rover and the planetary surface over which it is moving. This can be made available early on in the rover development cycle and can be used to feed information back into the rover design. It is also relatively easy to introduce faults into a simulation.

The University of Dundee, with funding from ESA, has developed a realistic planetary surface and sensor simulation facility for use in developing vision-based

*Director, Applied Computing. Member AIAA.

†Post-Doctoral Research Assistant, Applied Computing.

‡Research Lecturer, Applied Computing.

§Guidance and Navigation Control Engineer, ESA/ESTEC Technical Directorate - Control System Data Division.

navigation systems for planetary landers. This system goes a long way in providing the required rover simulation environment. This chapter describes the potential use of that simulation facility for training future planetary rover operators.

The Planet and Asteroid Natural Scene Generation Utility (PANGU) is a software tool for simulating and visualizing the surface of various planetary bodies. It was designed to support the development of planetary landers that use computer vision to navigate toward the surface and to avoid any obstacles near the landing site. PANGU can generate artificial surfaces representative of the moon, Mercury, Mars, asteroids, and comets, and provide images of the simulated surfaces. When given the position and orientation of a spacecraft above the planet's surface, PANGU responds by producing an image of the surface from that viewpoint. Recent research has extended the capabilities of PANGU to include scanning light detection and ranging (LIDAR) and RADAR altimeter sensors, providing a comprehensive framework for simulating navigation sensor responses during the descent of a lander. Research has also been done on simulating dust clouds on Mars.

PANGU may also be used to simulate a rover driving over the surface of a planet, following the terrain and sensing the environment around it. Multiple cameras can be simulated along with other hazard detection sensors like laser beams. When integrated with a rover control system simulation and telemetry delay simulation, PANGU can provide a realistic environment for training operators of surface exploration missions. To support integration with the simulations of other mission elements, PANGU uses Transmission Control Protocol/Internet Protocol (TCP/IP) as a basis for sending commands and receiving image data.

PANGU has been used by ESA, European Aeronautics Defence and Space Company (EADS) Astrium, Officine Galileo, National Institute of Engineering and Industrial Technology (Portugal) (INETI), and University of Dundee in the development of a navigation camera for planetary landers.

II. Aurora

The ESA Aurora program was given the green light at an ESA council meeting on 5–6 December 2005. Aurora received extensive interest from the ESA member states with 14 countries subscribing to the program. The Aurora program aims to explore planetary bodies within the solar system that may hold traces of life. The exploration will be done with robotic missions initially, but the intention is to extend this to human exploration in the future [1].

The first mission within the Aurora program is ExoMars, which will be launched in 2011. ExoMars will put a rover on the surface of Mars carrying an instrument suite designed to look for signs of life. Included on the highly mobile rover will be a drill that can reach up to 2 m below the surface of Mars [2].

Following ExoMars will be the Mars Sample Return mission, which will return samples of Martian rock and soil to Earth for analysis. This mission is highly ambitious, combining planetary landing, rover, ascent, rendezvous, return to Earth orbit, reentry, and recovery. ESA has been pursuing a program of technology development to assess and mitigate the risks involved in such a mission. Further essential technology development will be done in the frame of the Aurora program.

III. Operator Requirements

Mission operators will need extensive training to operate a rover on the surface of Mars and to be able to cope with problems that occur. This section considers some of the requirements for a computer-based rover and planet surface simulation system that would be of use for training mission operators.

The first consideration is the simulation of the Martian surface. This must be realistic, implying high-resolution surface models, and must be fairly extensive to give the rover sufficient room to roam. Both small- and large-scale features have to be simulated. Large-scale features can be modeled using existing Martian digital elevation models (DEMs), for example those produced by Mars Orbiter Laser Altimeter (MOLA) [3] or from the stereoscopic images collected from Mars Express [4]. Small-scale features like boulders, sand dunes, gullies, and craters have to be simulated based on known scientific information about the characteristics of these features. Realism is a key driver in the development of these models, especially when they may be used to produce image sensor data where any defects are clearly apparent and will have an effect on any image processing being performed.

The second set of requirements covers the rover simulation itself. The simulated rover must be fully representative of the rover for which the operators are being trained. It must respond to the same set of commands and must give the same set of status information. The navigation sensor suite that it carries must be the same, and cameras and other sensors must be located in the same positions as on the real vehicle. The rover must interact with the surface in the same way as the real vehicle. This includes resting on the underlying surface, exerting traction on the surface, and imitating the dynamic characteristics of the vehicle, in particular the suspension characteristics. The rover has to interact with objects like small boulders, either rising up over them, or in the case of larger boulders, avoiding collisions with them. A visual representation of the rover is also necessary where it appears in its sensors' fields of view, where a static lander is being used by mission operators to monitor the initial motion of the rover, or where two or more rovers are operating together. The rover may also leave tracks on the surface that it has traversed. Simulation of traction requires the surface model to hold information about the nature of the surface. The leaving of tracks necessitates some means of texturing the surface with a representation of the tracks.

The third set of requirements is related to sensors on the vehicle. These sensors may include cameras, laser range finders, LIDAR, tactile, and other navigation sensors. There may be one or more cameras on the rover, which may be firmly fixed to the rover frame with fixed fields of view, may be capable of pan, tilt, and zoom operations, or may be attached to a robotic arm. The camera simulation must allow specification of number of pixels and color or black and white operation. Laser range finders and LIDAR are similar types of laser-beam-based sensors. They may be used simply to detect obstacles in the front of the rover, to make distance measurements, or to scan the terrain, producing a three-dimensional representation of the surface in front of the rover. The most important characteristics of this type of instrument are the direction that it is pointing in, the minimum range, and the resolution. Other obstacle detection sensors, for example tactile sensors or tilt sensors, may be included on the rover and have to be modeled.

Finally there may be some other navigation sensors on the vehicle, for example inertial measurement sensors, that have to be simulated.

The next set of requirements covers communications of the rover with either a lander, or one or more orbiting spacecraft, which relay the information to and from Earth. The main tasks for the simulation are to simulate the availability of the communications link, only allowing communications when the rover and relay element are in line of sight, and to simulate the communications time delay, which can be significant for Earth to Mars communications.

Power availability for the rover also has to be simulated. Assuming that power is provided by batteries that are charged from solar panels, the orientation of the panels to the sun needs to be considered along with any shadowing from large boulders, etc. The level of charge in the batteries needs to be simulated so that the movement of the rover can be restricted according to the energy taken from the battery. Battery status will need to be simulated and reported to the operator along with other status information.

The simulation will require capabilities to help with evaluating mission operator performance. For example, an operator task may be to arrange for the rover to reach a specific target feature on the surface. Useful measurements made by the simulation system should include the final distance from target, the route taken to the target, the optimum route from starting point to target, the energy used to reach target, and the level of hazards encountered en route to the target.

The user interface to the simulation system has two sets of requirements: one relating to the operator interface and the other relating to configuration of the planet surface simulation, rover, sensors, and fault injection facilities. The operator must be able to enter commands to the simulation and receive status, science measurements, navigation information, and camera images, in the same way as the real rover would be controlled. The configuration information must give access to all of the configuration parameters of the simulation, allowing planet surfaces to be constructed easily from existing DEMs and small-scale feature models, rovers to be configured with different physical characteristics and different sensor packages, and faults to be injected into the simulation.

There are a number of requirements related to fault simulation that have to be covered. Communication with the rover can suffer possible problems, for example the temporary loss of communications with an orbiting relay satellite, reducing the available time for communicating with the rover. The operator response to this type of situation is an important area for training. Situations like the loss of communication in one direction only, where the rover can be commanded to enter a safe mode positioned to optimize signal reception while attempts are made to restore communications, may be required. There may be problems caused by erroneous autonomous operation of the rover, where it has taken a different route to that expected during a routine communications break. When communications are reestablished, the operator has to determine where the rover actually is so that it can be redirected back toward the target. The rover itself may suffer a partial fault, for example loss of motive power to a wheel. The operator has to determine if this is a mission critical failure and if there is anything that can be done to mitigate the effects of the fault. The rover may run into a rock if one of its obstacle detection sensors has failed. In this case the rover may stop waiting for help from the operator. The operator must then quickly assess the situation and send

appropriate commands to the rover to help it recover from the problem. The rover may also become stuck when there is no sensor fault, for example on a sandy incline where the rover continually slips down a slope it is trying to climb. The operator has to find an alternative route, avoiding the sandy slope. The rover may suffer loss of one of its solar panels, reducing the power available for operation. The operator must then consider operations that maximize the exposure of the working solar panels to the sun. Recovery from a temporary complete power failure may also need to be rehearsed.

Another set of requirements for the simulation system covers its use in the testing of any autonomous navigation algorithms. Cameras may be used to detect obstacles on the required route and to help with navigation. The simulation should support the development of the related hardware and software by providing near real-time sensor simulation, with software and hardware in-the-loop control of the rover simulation. The testing of the interaction between the mission operator and the rover autonomy is also an important consideration in the design of the simulation system.

The final set of requirements is concerned with performance. Realistic (high-resolution), large-scale surface models contain a large amount of data that can take a long time to process. Human in-the-loop operation requires response times that are fairly fast, although the Earth-Mars-Earth time delay may allow some leeway here. The testing of autonomous control systems requires at least near real-time performance. These high performance requirements and high level of realism give rise to opposing constraints on the system and a tradeoff has to be made.

IV. PANGU Simulation Tool

PANGU is a software tool for simulating and visualizing the surface of various planetary bodies [5–13]. It has been designed to support the development of planetary landers that use computer vision to navigate toward the surface and to avoid any obstacles near the landing site. PANGU can be used to generate an artificial surface representative of the moon, Mercury, Mars, or asteroids and to provide images of the simulated planetary body. When given the position and orientation of a spacecraft above the planet's surface, PANGU responds by producing an image of the surface from that viewpoint. Recent developments of PANGU have included a scanning LIDAR simulation [10] and a RADAR altimeter simulation [13]. These simulations allow comprehensive simulation and evaluation of various navigation sensor systems for planetary landers. An example PANGU image of an asteroid and a lunar-like surface are shown in Fig. 1.

The architecture of the PANGU software is illustrated in Fig. 2. Rectangles represent the programs that make up PANGU, cylinders show the parameter and data files, and the arrows illustrate the flow of information through the system.

The surface generator is responsible for generating a realistic planet surface based on parametric representation of the required surface. It uses quantitative measures meaningful to planetary scientists, for example crater-size density distribution or crater aging characteristics. The surface generator takes information from the surface parameter file, which determines the size of the surface to be generated and its roughness, etc., and builds a basic undulating surface devoid of

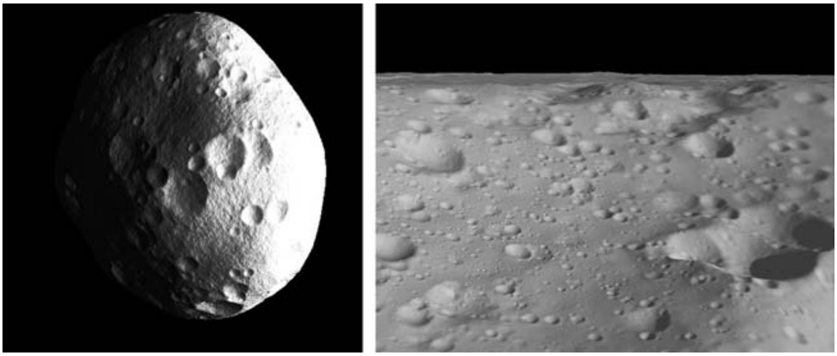


Fig. 1 Asteroid and lunar surface simulation with PANGU.

surface features. To this surface various types of feature, for example craters and boulders, are added. The parameters for each feature, for example position, size, and age, are contained in a feature list file. There is one feature list for each type of feature to be added to the surface. The models for the various features are defined in feature model files. The feature models are based on idealized scientific knowledge of expected features on the surface. These idealized models are made to look realistic using fractal techniques. There is one feature model file for each type of feature being added to the surface. With this information, features are added to the surface by the surface generator.

The surface generator produces a DEM for the surface, with craters and other features that result in a single height point for each position on the surface (monotonic features). From the DEM it then produces a polygon representation of the surface to which it adds boulders. Boulders may result in three height points for

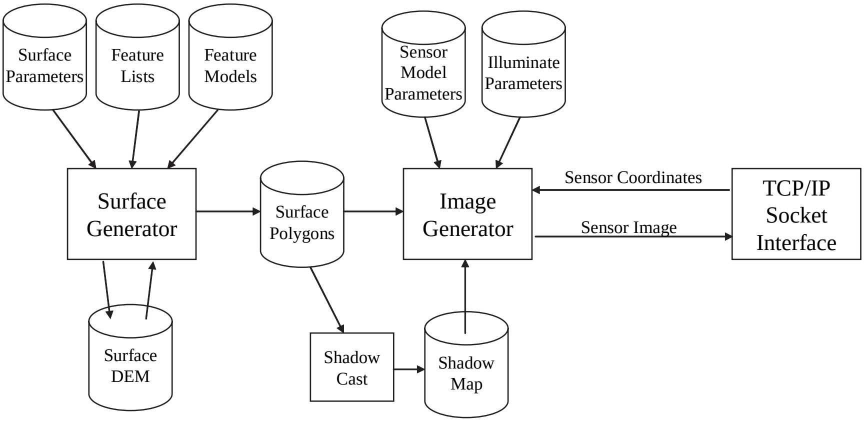


Fig. 2 PANGU software architecture.

any point in the surface, when part of the boulder "overhangs" its base, which is why they are added to the polygon model rather than the DEM.

The surface polygon model is used to generate a shadow map of the surface that identifies which polygons or facets of the surface model are in shadow. Both surface generation and shadow casting are computationally intense tasks, and so they are performed offline before the surface is visualized. The shadow map is only needed for passive optical instruments: LIDAR instrument and RADAR altimeter simulations do not need a shadow map.

With the surface polygon model and shadow map prepared, images or other sensor data of the surface can be generated. The image generator requires information about the sensor being used to produce an image, for example number of pixels and field of view, and also about the illumination conditions including sun orientation and intensity. This information is provided by the sensor model and illumination parameter files. The image generator is controlled via a TCP/IP interface. The PANGU user sends a vector containing sensor position and orientation relative to the surface, and the image generator responds by rending the appropriate image of the surface, taking into account the sensor and illumination parameters. The image is returned to the user over the TCP/IP interface.

The time taken to render an image depends on the size of the image and the number of polygons in the surface model. Typically, on a recent PC, this takes less than 1 s even for large surface models and 1×1 k images. This level of performance enables near real-time image generation and the use of PANGU in a simulation loop with real hardware and/or software. LIDAR and RADAR altimeter simulations are also fast.

PANGU is easy to connect to another system, for example an image processing system or Guidance and Navigation Control (GNC) simulation. The TCP/IP socket interface to PANGU enables any Internet enabled computer to connect to the PANGU tool, control it, and receive images from the surface simulation. The image generator component of PANGU acts as a server, listening for a connection to be established with another machine. A client running on the other system opens a TCP/IP socket connection with the PANGU server and transmits the sensor coordinates over the connected socket. The PANGU server then generates the image and sends it back over the socket connection to the client system. If multiple sensors are to be included in a simulation, then they can be run on separate PCs to improve performance.

V. Planetary Lander Operations

Planetary landers can use computer vision in several different ways: surface relative navigation, determining the motion of a lander relative to the surface; transverse velocity measurement, necessary so that any residual transverse velocity can be nulled before landing to prevent the lander from rolling over; and hazard detection to assist with possible avoidance of obstacles on the surface in the vicinity of the landing site.

Surface relative navigation requires a sensor that can pick out features or landmarks on the surface and use these to track the position of the spacecraft relative to the surface. There are two techniques that are currently being developed: passive and active vision.

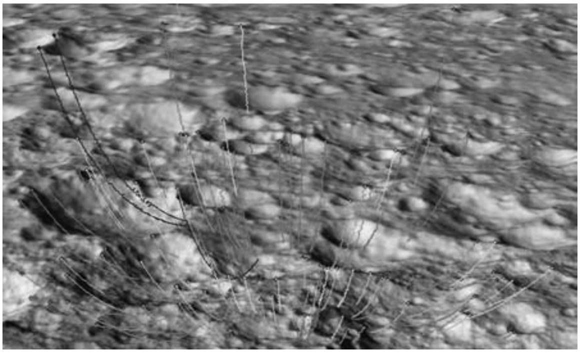


Fig. 3 Tracking features in a PANGU image sequence. (See also the color figure section starting on p. 645.)

Passive vision uses a camera to record images of the surface during descent. Features are picked out on the images and tracked from frame to frame. The motion of the spacecraft relative to the surface is then determined from the tracks of these image features along with some other navigation information (Fig. 3). Passive vision requires a well-illuminated target surface on which to land. Landing on the dark side of a planet, or on planetary bodies a long way from the sun, is not possible with passive vision systems unless an illumination source is carried on the spacecraft.

Active vision provides illumination of the surface to support optical measurements. LIDAR is the prime example of an active vision system. A pulse from a laser illuminates a point on the surface. Reflected light from the surface point is received by the LIDAR receiver and used to measure the distance to that surface point. Active vision systems can be used for landing on both sunlit and dark sides of planets, for landings in heavily shadowed areas, and for missions landing on planetary bodies distant from the sun. LIDAR vision can also be used as a complementary and redundant system to a passive vision system.

Transverse velocity measurement can also be performed using feature tracing with a passive vision system. Combined with information from the inertial measurement unit and/or a RADAR altimeter, it is relatively straightforward to produce an estimate of the transverse velocity of the lander from an image sequence.

As the lander approaches the target landing spot, it can detect obstacles in the vicinity of the landing spot that were not visible from an orbital survey. Obstacle or hazard detection processing on information from a camera or LIDAR can be used to help pilot the spacecraft to a safe landing spot.

PANGU has already been used to help develop and test each of these vision-based navigation and hazard detection techniques.

VI. Rover Operations

Martian surface simulation has been incorporated in PANGU with the inclusion of suitable boulder models, and more recently, work on sand dune simulation.

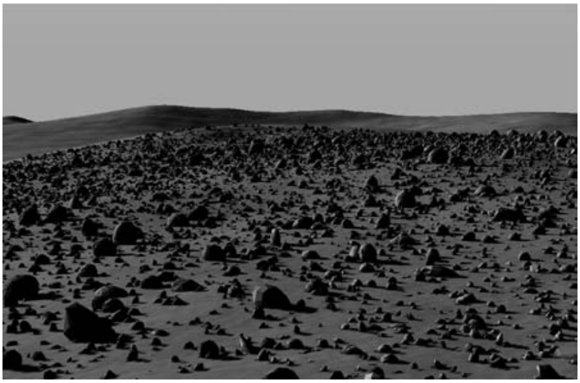


Fig. 4 Martian surface simulation.

An example image of a PANGU Martian surface is shown in Fig. 4. The underlying terrain was constructed from MOLA data, which provides one DEM point every few hundred meters. This low-resolution grid was interpolated using fractal techniques to produce the surface seen in Fig. 4. Boulders were then added to this surface. The image was taken with a simulated camera a short distance above the surface.

An initial prototype of a rover moving over the surface has been produced. The simulated vehicle follows the surface as it moves across it. Several cameras may be simulated, fixed at different locations on the rover with different viewing directions. The architecture of PANGU allows many concurrent sensor simulations to be run on different computers, to speed up the simulation. Each simulation is connected using a TCP/IP socket and given the position and orientation of the camera for the next frame. PANGU then produces the appropriate image. A central computer computes the rover position and orientation on the surface and then computes the position and orientation of each sensor. The sensor position and orientation are then sent out to the individual sensor simulation computers.

VII. Does PANGU Meet the Operator Training Requirements?

It is worth considering to what extent the existing PANGU tool meets the requirements of the Martian rover simulation listed in Sec. III and what requirements need extensions or modifications to PANGU:

1) Simulation of Martian surface: PANGU is capable of simulating a range of different planetary surfaces. Martian-type surfaces with boulders and Martian profile craters can be simulated, and work is ongoing on the simulation of sand dunes. Because of the recent extensive large-scale mapping of Mars by MOLA and Mars Express, which have produced DEMs of all or part of the Martian surface, a tool has been added to PANGU to enable existing DEMs to be used as the basis for a Martian surface model. The large-scale DEM is fractally interpolated, and small-scale features are then added to the surface. Research is planned to further enhance Martian surface modeling within PANGU.

2) Rover simulation: The principal requirements here are dynamic modeling of the rover, interaction with the surface and small objects, and visual representation of the rover. A program of work is currently under way, initially concentrating on being able to add CAD models of rovers and spacecraft into PANGU, and then looking at dynamic modeling and surface interaction. At present it is possible to add spacecraft models designed using a variety of CAD tools into PANGU.

3) Sensors: Cameras, laser range finders, and LIDAR can all be simulated with PANGU at present. Tactile and inertial navigation sensors are relatively straightforward to simulate but are not yet included in the PANGU toolset.

4) Communications: Rover to Earth communication simulation is not included in the current PANGU toolset. Ray tracing could be used to assess whether there is a line of sight between transmitter and receiver.

5) Power: The assessment of available rover power is not included in the current PANGU toolset. PANGU could support measurement of the power being generated by the solar panels, based on the sun orientation relative to the solar panel and the use of shadow maps or ray tracing.

6) Operator measurements: Measurements of actual rover position, distance from the target, etc., can be readily added to PANGU. The inclusion of software to determine the optimum route to a target is beyond the scope of the current work on PANGU.

7) User interface: Configuration of the planet surface simulation and sensors is currently done using the various parameter files. A graphical user interface is planned to help with the design of planetary surfaces. The user interface to the rover will be designed as part of the current work on rover simulation. There are currently no other user interfaces to PANGU. The design philosophy behind PANGU was to let the end users develop their own user interface on a separate PC communicating with PANGU via TCP/IP. A rover operator user interface will have to be developed for PANGU.

8) Fault simulation: This is not currently included in PANGU.

9) Near real-time sensor simulation: PANGU supports near real-time sensor simulations.

10) Realism: PANGU does produce realistic, large-scale planetary surface models. Further work is required on the Martian surfaces, and this is currently under way.

Overall PANGU is well placed to fill the requirements listed in Sec. III; the planet surface and sensor simulation is ready with only a few enhancements required. The rover simulation needs to be implemented, but already basic spacecraft models can be incorporated in PANGU. The requirements specific to operator training including operator interface, communications delay, and fault injection need to be done.

VIII. Testing Autonomous Control Systems

As well as providing a platform for operator training, PANGU would also be suitable for the testing of autonomous rover navigation and hazard detection techniques. This is a relatively straightforward extension of the planetary lander simulation provided by PANGU. This would provide a calibrated simulation environment for testing candidate computer vision methods that would be used alongside physical simulation environments.

IX. Conclusion

This chapter has examined the requirements for a computer-based simulation system for training the operators of Mars rovers. It has introduced the PANGU planet surface simulation tool, described how this tool has been used to support the development and testing of vision-based navigation systems for planetary landers, and explained how the tool is being developed to support rover simulation.

PANGU provides an important facility to support the ESA Aurora program for the testing of vision-based navigation systems and hazard detection systems for planetary landers, for testing rover autonomous navigation and obstacle detection systems, for testing sample-return canister in-orbit rendezvous systems, and for training mission operators, especially those responsible for rover operations.

PANGU can produce simulations of asteroids, and so it could also be used for operator training for asteroid or comet rendezvous missions or for missions to the moons of Mars.

Current work at the University of Dundee is concentrating on calibrating the PANGU camera simulation, providing sand dunes, performing transverse velocity measurement, and extending the PANGU tool to implement a broader set of the requirements listed in Sec. III.

Acknowledgments

The authors from the University of Dundee thank the ESA for financial support of the PANGU tool development, which was done under ESA contracts 17338/NL/LvH/bj and 11747/95/NL/JG.

References

[1] “Aurora Exploration Program,” ESA, http://www.esa.int/esaMI/Aurora/SEMZOS39ZAD_0.html.

[2] ExoMars Web Site, ESA, http://www.esa.int/esaMI/Aurora/SEM1NVZKQAD_0.html.

[3] MOLA Web Site, NASA, http://ltpwww.gsfc.nasa.gov/tharsis/mola.html.

[4] Mars Express Web Site, ESA, http://www.esa.int/SPECIALS/Mars_Express/index.html.

[5] Parkes, S. M., Martin, I., and Milne, I., “Lunar Surface Simulation – Modelling for Vision Guided Lunar Landers,” *Proceedings, DASIA (Data Systems In Aerospace) 99*, ESA SP-447, Lisbon, Portugal, 17–21 May 1999, pp. 369–374.

[6] Parkes, S. M., and Martin, I., “Virtual Lunar Landscapes for Testing Vision-Guided Lunar Landers,” *Proceedings, 1999 IEEE International Conference on Information Visualization*, 14–16 July 1999, London, pp. 122–127.

[7] Parkes, S. M., and Martin, I., “Virtual Reality for Real Spaceships,” *Proceedings of International Conference on Virtual Systems and MultiMedia*, 1–3 Sept. 1999, Dundee, Scotland, UK, pp. 156–168.

[8] Parkes, S. M., and Martin, I., “Lunar Surface Simulation – Opening the Road to the Moon,” *Proceedings of the Fourth International Conference on Exploration and Utilisation of the Moon*, ESA SP-462, July 2000, pp. 221–224.

[9] Parkes, S. M., Martin, I., and Strandmoe, S., “Planet Surface Simulation for BEPI Colombo,” *DASIA 2001: Data Systems in Aerospace – Proceedings of an International Symposium*, ESA SP-483, Nice, France, 28 May–1 June, 2001.

[10] Parkes, S. M., Martin, I., Dunstan, M., and Mills, S., "Mercury Surface Simulation for Bepi Colombo Lander," *Proceedings, DASIA 2002*, ESA SP-509, May 2002, pp. 1–7.
[11] Parkes, S. M., Dunstan, M., Matthews, D., Martin, I., and Silva, V., "LIDAR-Based GNC for Planetary Landing: Simulation with PANGU," *Proceedings, DASIA 2003*, ESA SP-532 [CD-ROM], June 2003.
[12] Parkes, S. M., Martin, I., and Dunstan, M., "Planet Surface Simulation with PANGU," *Proceedings, Eighth International Conference on Space Operations* [CD ROM]; also http://spaceops2004.org, pp. 1–10.
[13] Parkes, S. M., Dunstan, M., Martin, I., and Mancuso, S., "RADAR Altimeter Simulation with PANGU," *Proceedings, DASIA 2005*, ESA SP-602 [CD-ROM], May–June 2005.

Chapter 33

Cassini–Huygens Sequence Development Process

Jennifer Long Maxwell,* William M. Heventhal III,† and Shahram Javidnia*

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

I. Introduction

THE Cassini spacecraft consists of 12 instruments: four optical remote sensing (ORS) instruments, six in-situ observation instruments to study magnetosphere and plasma science (MAPS), one RADAR instrument, and one radio science (RSS) instrument. When this complex mission was initially architected, much of the early emphasis was placed on the spacecraft function and design, rather than operations. The spacecraft and mission design posed significant challenges to the science and sequence development process for the four-year tour of the Saturnian system.

In Cassini–Huygens mission operations, a sequence refers to one or more programs containing both instrument and spacecraft commands that will execute for a desired amount of time. During Cassini's prime tour of Saturn, each sequence lasts for approximately 40 days. By developing the spacecraft activity plans ahead of time, operations costs are reduced because the engineers and scientists can work on other tasks rather than interacting with the spacecraft on a daily basis, which can be slowed by the light time delay when sending commands to Saturn. This approach to spacecraft commanding also has an advantage with a spacecraft as complex as Cassini because it provides a single, integrated product that can be constraint checked and tested to ensure all activities are coordinated. The major disadvantage with this approach, however, is the lack of flexibility to major sequence changes as execution approaches. This is primarily a factor when dealing with changes to the Deep Space Network (DSN) coverage that may come in

---

*Sequence Virtual Team Lead, Cassini Science and Uplink Office. Member AIAA.

†Uplink Operations Manager, Cassini Science and Uplink Office.

Table 1. Sequence development timeline

When	What (goals)
10 years before prime mission (PM)	Tour design (maximize science opportunity)
4 years before PM	Integration (negotiate best science compromise)
2 years before PM	Science operations plan (SOP) implementation (validate basic sequence design)
20 weeks before execution	Aftermarket (update integrated plan)
15 weeks before execution	SOP update (update basic sequence design)
10 weeks before execution	Science and sequence update process (SSUP) sequencing (validate entire sequence)

late in Cassini's sequence development process due to coordination with other spacecraft missions.

The development of a sequence is a coordinated effort led by two teams: science planning and uplink operations (ULO). A sequence can begin the initial planning stages as early as 10 years before prime mission yet is not finalized until one week before execution. Table 1 illustrates a general timeline of each phase of sequence development for the Cassini mission.

The end-to-end sequence design process consists of five phases:

1) Integration of the science operations plan (SOP), a high-level plan of science and engineering activities, detailing their timing, power, thermal, data volume, and pointing profiles.

2) SOP implementation, in which resource conflicts are resolved and activities constraint checked.

3) Aftermarket and SOP update, in which the SOP is updated while in tour using the latest information on the navigation ephemeris, and the spacecraft's and instruments' performance.

4) Science and sequence update process, which results in an integrated, validated, uplinkable, and flyable distributed sequence.

5) Execution, which includes system-level and instrument-internal real-time commands, anomaly response, and sequence pointing and timing adaptation using the latest ephemeris information.

The science planning process was created from a series of mission planning requirements and models from past planetary missions, including the program documents, *Cassini Operations System Functional Design* and *Cassini Operations System Functional Requirements*. Driving requirements were identified as accepted, modified, or rejected, and processes or tools were assigned to each requirement. From these basic guidelines and constraints, a science planning process was created that consists of the integration, implementation, aftermarket, and update of the SOP.

After the science planning process is complete, the products are handed off to ULO, which leads final verification and detailed command generation during the science and sequence update process (SSUP). ULO then uplinks the commands to the spacecraft, and supervises the sequence commands as they execute. Any real-time engineering and science activities are coordinated through ULO during execution.

Each phase of the sequence development process had to overcome many operational challenges because of the immense complexity of the spacecraft, tour design, pointing capabilities, flight rules, and software development. This chapter will address the specific challenges related to each of those complexities and the methods used to overcome them during operations.

II. Science Planning Process

A. Activity Generation, Integration, and Conflict Resolution

Figure 1 shows the flow of the activity generation process for tour into the Cassini Information Management Systems (CIMS) [1]. CIMS is the database where observation request details such as pointing profiles, data volume, power requirements, telemetry modes, and timing are stored.

The science teams are responsible for prioritizing and developing conflict-free activity requests and plans for their experiment objectives throughout tour. The Spacecraft Office (SCO) and the Instrument Operations Team (IO) have to deliver all requests for engineering-related activities needed in the tour. These requests are input into CIMS for use during the science planning integration and implementation processes.

The Discipline Working Groups (DWG) were established to take into account the main science objectives of the mission and identify methods throughout the tour to attain these goals. These working groups were the Rings Working Group, Atmospheres Working Group, the Magnetosphere and Plasma Science Working Group (MAPS), and Saturn Working Group. Once this task was completed, the

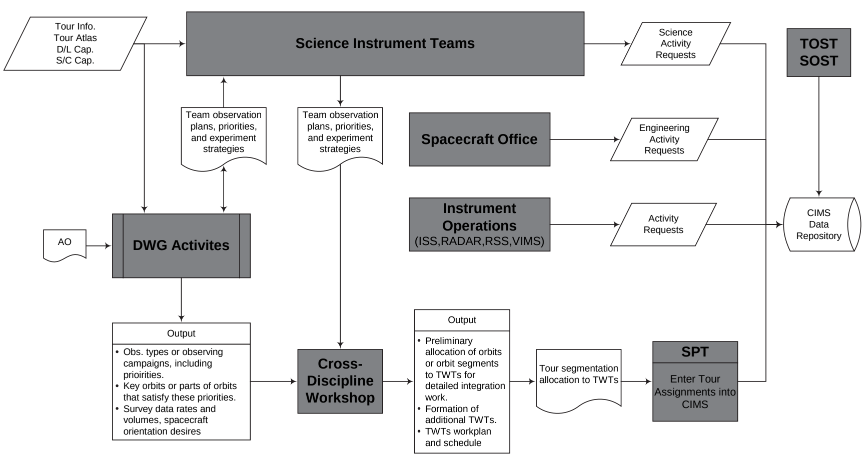


Fig. 1 Activity request generation process flow.

tour segments were assigned to different Target Working Teams (TWT) and Orbital Science Teams (OST) for detailed integration.

The groups responsible for integration are the Titan Orbiter Science Team (TOST), Satellite Orbiter Science Team (SOST), and the Target Working Teams (TWT), divided by discipline into four groups: Cross-Discipline, Rings, Saturn, and Magnetospheres. Integration is the process in which the science and engineering activities are incorporated into a coherent timeline within each TWT/OST. The science planning engineer aides this process by creating detailed activity reports from the information in CIMS and proposing solutions to any conflicts. The science and engineering teams resolve the conflicts during the TWT/OST process. Once each TWT/OST completes a timeline, the products are archived and later merged to create a single activity plan for the implementation process.

B. Implementation

Implementation is the process in which integrated activity plans are constraint checked and turned into spacecraft command modules that will later be expanded and radiated to the spacecraft. Members of the science planning (SP) team, SCO, IO, ULO, mission planning, and the science instrument teams make up the Science Planning Virtual Team (SPVT) that performs this process. This was primarily implemented as a two-phase process. The main milestone in each phase is referred to as a port in which the instrument team commands are merged into an integrated sequence that is constraint checked to ensure the spacecraft remains safe. The two-port process design was based on estimated changes made during implementation and the amount of time needed to accommodate them. For the beginning sequences, this process was implemented as a three-port process and later reduced to two as the learning curve stabilized. At the end of each implementation process, the sequence history and liens are documented, and the instrument and merged sequence spacecraft activity and sequencing files (SASF) are archived in the project database [Distributed Object Manager (DOM)].

C. Aftermarket and SOP Update

The aftermarket process occurs when science and engineering activities are updated for new discoveries, DSN changes, or modifications to the tour design. During this process, each team submits its requested changes to the SPVT Lead. These changes are evaluated based on the estimated work hours needed for implementation. If the work hours to implement all changes are below the limit set for that sequence, then the changes are allowed to proceed to reintegration. If there are more changes requested than work hours allocated for that sequence, negotiations are performed to reduce the amount of changes implemented for that sequence. There is a reintegration period during which the TWT/OST manages the CIMS database to produce new products for the SOP update process. A wrap-up meeting is scheduled for the handoff of those products to the start of SOP update.

Once this handoff takes place, the five-week SOP update process begins. This process parallels one port of the implementation phase. During this time, updates to the pointing designs and data policing tables, which control the amount of data volume instrument teams are allocated at a given time, take place. In addition, the

latest ephemeredes and DSN station allocations are incorporated into the sequence. The sequence is constraint checked and the pointing profile is validated. At the end of SOP update, a conflict-free, integrated SASF containing the basic instrument and subsystem commands is handed off to the ULO for final detailed command development.

III. Science and Sequence Update Process

The SSUP consists of four phases: subsequence generation (SSG), sequence integration, review, and validation (SIV), real-time command preparation, and sequence and command radiation. Because the sequence commands are in their final stages of validation before uplink to the spacecraft, SSUP employs more stringent configuration management through its policies and procedures. Figure 2 details the steps of the SSUP process [2]. The development team during this process is known as the Sequence Virtual Team (SVT) and consists of many of the same instrument, spacecraft, and mission planning team members involved in the SPVT, along with members of the mission control team to coordinate the uplink and real-time activities.

A. Subsequence Generation

During the SSG phase, the SOP update products are used to create team-stripped SASFs. By providing teams with the baseline product that has already been constraint checked, the possibility of introducing additional errors is minimized. Teams review these products and fill in any empty requests that were provided as placeholders during earlier development processes. Existing activities are reviewed for desired design changes or errors identified during the SOP update process. Changes to existing designs that affect a system-level resource must be identified

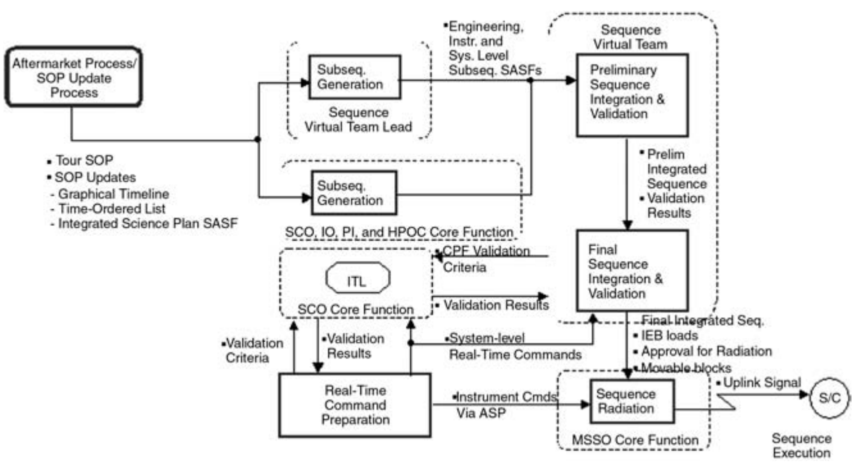


Fig. 2 SSUP process.

with a sequence change request (SCR) that is reviewed and approved by the SVT. Once approved, these changes can be submitted in the SASF along with other accepted changes to triggers and other team internal resources.

B. Sequence Integration, Review, and Validation

The SIV process consists of a preliminary and final phase, with the preliminary phase including two cycles and the final phase including only one cycle. During the preliminary SIV phases, a merged product is created from the science instrument and engineering SASFs submitted at the end of the previous phase. A movable block is a pre-identified portion of the sequence that may be updated in time during a later phase. If this exists in the sequence, it is extracted and developed as its own product, as well as merged as part of the background sequence to ensure compatibility. The merged sequence products are reviewed by the instrument teams and by the spacecraft office for any conflicts or flight rule violations. The vector commands are generated from the attitude control team and merged with the background sequence to create a complete product that could be uplinked to the spacecraft. Changes as a result of the review of these products must be identified through SCRs for SVT review and approved by the SVT lead (SVTL). Once approved, these changes are delivered in the SASF for the following phase. Changes for the final SIV phase are allowed for health and safety reasons only. First-time events or special activities within the sequence are also tested using the Integration Test Lab (ITL). Instrument expanded blocks (IEBs), subroutines stored in instrument memory for sequence activities, are delivered and processed during this phase. The SIV phase results in a conflict-free, integrated, uplinkable distributed sequence.

C. Real-Time Activities

Real-time activities are prepared before and during sequence execution. Spacecraft resource users, both instrument teams and the spacecraft office, submit their real-time activities in an SASF. The SVTL adds commands to the SASF for memory management and performs constraint checks. If the real-time activity is system level, the appropriate teams review and approve the file before uplink. For instrument internal commands, they can be coordinated through this process, if memory management is needed, or sent via the automated sequence processor. This processor will autonomously build the commands that are sent directly to the instrument and submit them for radiation over the next uplink opportunity without involvement of the SVT.

D. Sequence and Command Radiation

The sequence and command radiation process prepares the DSN complexes for radiation of background sequence and real-time commands. The process requires interaction between two teams that authorize the uplink of files to the spacecraft. Once the DSN station has acquired the signal with the Cassini spacecraft, telemetry is received and initial conditions are verified, if necessary. The SVTL or spacecraft engineer verifies the command to be uplinked by confirming the time the file was

created and the name of the file. These two fields provide unique characteristics for each command file to ensure the correct files are uplinked to the spacecraft. The mission controller verifies the file upon authorization from the SVTL or spacecraft engineer and places it in the queue for radiation. This file is again verified by the SVTL or spacecraft engineer, and a "go" is given to the mission controller for radiation of that file either immediately or at a specified time during the uplink opportunity. Once the file is active, the command packets are transferred to the DSN complex and radiated to the spacecraft. Verification of the file is done through registration of the program in sequence memory or execution of the commands seen by the instrument team or spacecraft subsystem.

IV. Lessons Learned

A. Web Management of Distributed Operations

1. Background of Distributed Operations Design

The Cassini–Huygens mission to Saturn and Titan is a distributed operations (DO) project. DO was chosen in an effort to reduce cost and maximize expertise [3]. Because Cassini–Huygens was conceived as an international collaboration, it was decided that it was more cost effective not to relocate the experts of the international teams to the Jet Propulsion Laboratory (JPL) for the duration of the mission. This also allowed for better suited instrument operating teams because the experts in each scientific field could be members of the team, and the teams could better tailor their products (science data) to the end-user community.

2. Using Technology to Create a Virtual Operations Team

The concept of a science planning and sequence virtual team was created to allow a core group to handle the sequence design and development processes. The use of mediums such as e-mail, web pages, and teleconferencing were essential to the implementation of the DO concept for the Cassini–Huygens mission. In addition, other tools and resources were developed to overcome the challenge of not having the core team of sequence developers working in a centralized location.

3. Design Tool and Information Distribution to JPL Teams and Remote Sites

Because of the DO system, one of the challenges was to effectively coordinate the software and development tools with the DO sites. Export regulations added a layer of complexity in distributing software to our international partners. The use of the Internet was crucial in this effort, with the creation of downloadable design software and centralized information databases. The science planning process utilized the web to ease the interface with the distributed teams with the CIMS database. This was the database in which science planning manages the overall pointing designs, telemetry modes, power modes, data volume allocation, and timing resources for the SOP. By making this a web-based interface, all members of the planning team could refer to the same resource in a timely manner. In addition, the real-time command generation process utilized a web-based command

approval request form through which the distributed teams could initiate or provide approval to their commands for uplink.

Providing configuration management and distributing configuration files for the sequence design software tools was another challenge. The science planning and sequence team leads became the custodians for the ancillary files used during each sequence. The Sequence Phase List of Ancillary Files is the central listing of those files that is published to the DOM. To ease the interface for the DO teams, the list is also collected into a downloadable tar bundle for use with the download version of the pointing design tool (PDT). In addition, a system was developed to mirror the ancillary file directory structure created at JPL, so that distributed teams could download the configuration files and utilize the same ancillary file locations that were available to local users at JPL. The introduction of web-based solutions such as these provided significant time and money savings to the operations costs on the Cassini–Huygens mission.

B. Science Observations

1. *Pointing Design*

Early in the design of the Cassini spacecraft, a descoping effort eliminated the high-precision scan platform and the turntable for the science instruments. The instruments were then to be fixed-mounted to the base body on two pallets. While the elimination of these elements did simplify the structural and thermal design by deleting deployments and eliminating the sunshade, whose function would be handled by the high-gain antenna (HGA), it also eliminated the decoupled pointing of the instruments and the HGA. This meant that the ORS instruments would have to acquire data without the HGA pointed at Earth [4]. This complicated the pointing strategy and science trades that had to be made during the planning process.

As a reaction to this, every science opportunity could no longer be optimized for all science objectives. Titan flybys, for instance, were divided up among prime science instruments. One flyby would be a radio science (RSS) prime flyby for an occultation, whereas the next flyby may be for the ion neutral mass spectrometer to take measurements of the Titan atmosphere. In addition, attitude strategy spreadsheets were employed during the science planning process to manage pointing for each instrument. Prime pointing instruments had to accommodate riding instruments on observations, which increased the amount of coordination needed for pointing designs and changes. The science planning process coordinated this by creating a merged pointing profile of all the prime instrument requests prior to the official input port during SOP implementation and SOP update. During SSUP, changes to prime pointing designs require an SCR, which the riding instruments review and indicate any impact to their commands during the approval process.

Another challenge that had to be overcome because of the loss of the scan platform and turntable was maintaining the safety of the overall spacecraft pointing profile, while allowing individual instruments to take control of the pointing for their observations. A waypoint strategy was developed in which a safe attitude was chosen for a given period of time. Science observations had to turn to and

from that waypoint attitude to facilitate coordination between science activities. If a design is not conflict free by the end of the sequence development process, it can be removed from the sequence, and the spacecraft will remain at the designated waypoint during that time. While this strategy does not optimize science observation time, it does prevent the spacecraft from pointing in an unsafe orientation.

2. Trajectory Changes

Another challenge during the sequence design process was creating pointing profiles that were robust to trajectory changes so that they could be updated based on the latest ephemeris information. Those challenges were overcome through the use of turn margin, movable blocks, and pointing updates during sequence execution. The pointing uncertainties for the reference trajectory used in sequence development were analyzed to estimate turn margin needed during pointing activities. This turn margin is held in pointing designs until late in SSUP, when the last chance to accept an update to the reference trajectory has passed.

For science opportunities that are most sensitive to changes in pointing and rely on the latest trajectory information, movable blocks are utilized. There are two types of movable blocks: ground and live. Ground movable blocks are shifted in time when a new reference trajectory is incorporated during sequence development. Live moveable blocks are shifted based on the latest trajectory information that is available during sequence execution. These time shifts are performed by relating all science observations to an epoch time that can be updated in the CIMS database and incorporated into the sequence products without redesign of each observation.

One final method to respond to late changes in the ephemeris information is the live update process. The live update process can include the live movable block time shift or can be an update to only the vector definitions. During this process, the latest trajectory information available during sequence execution is analyzed and run through the pointing software. The inertial vector definitions are updated and overlaid onto the existing definitions that are onboard the spacecraft for a given time period.

C. Software Development

1. Flight Rule Checking

The Cassini spacecraft is a complex orbiter that requires detailed flight software (FSW) and flight rules to maintain the safety and operation of the spacecraft. Cassini has defined over 330 flight rules as a method of issuing spacecraft and instrument health and safety. Violation of these rules could threaten health and safety of an instrument or spacecraft subsystem. Many of those flight rules were coded into ground software (GSW) to minimize the likelihood of oversight that could damage an instrument.

2. Planned FSW/GSW Delivery

Another method to lower development costs prelaunch was to defer software capabilities until closer to the need date. For example, reaction wheel control

would not be needed for the first four years of flight. Consequently, Cassini launched without this capability and would later provide it as the need date arose. This provided some advantages to the flight software development and flight rule checking because it allowed for a better understanding of the spacecraft capabilities due to its in-flight performance.

A disadvantage to the delay in ground software capabilities, however, was reflected in early sequence development [4]. During the initial stages of activity planning and sequence integration, much of the needed planning software was not available to coordinate the activities planned for tour. Much of this analysis had to be done by hand or using software developed by the science planning team rather than by the software development team. This resulted in non-uniform products handed off from the early stages of sequence development. Once the planning software was introduced a few years after the planning effort began, a rework effort was required to integrate these early plans into the standardized templates. In addition, much of the software capabilities were developed very close to the need date, which resulted in some delay in the planning schedule.

3. *Multimission Software*

In an effort to lower software development costs, Cassini decided to take advantage of already developed, multimission software. Many algorithms needed are not dependent on any specific spacecraft mission. Consequently, these algorithms can be developed once and have this development cost spread over all missions who utilize this multimission concept. Because of the complexity of the Cassini spacecraft and operations, many of these tools had to be modified, which offset the cost savings provided by multimission software.

4. *High Speed Simulator (HSS)*

A high-speed simulator (HSS) was originally designed and developed as a less costly, less labor intensive and faster alternative to the ITL. The initial HSS concept was a multimission software simulator that employed the actual attitude and articulation control (AACS) and command and data system (CDS) FSW to mimic the sequence execution onboard spacecraft. The concept of a high-speed AACS and CDS software simulator proved to be too costly to implement, mainly due to the complexity of the AACS FSW and its interaction with CDS FSW. An additional challenge was to find affordable hardware that could execute the sequence in a timely manner, to conform to the SSUP schedule. The AACS FSW simulation requirement was eventually dropped in the Cassini-Huygens mission (and its predecessor Galileo), and the HSS was strictly utilized for the CDS capabilities. The Cassini AACS team developed a FSW simulator to perform a similar function.

Advantages of HSS over ITL are numerous, including lesser requirement of expertise for operations, faster than real-time execution, availability via any workstation (vs in the ITL location), ease of setup, no need for continuous personnel support, and lack of hardware configuration requirement. Given the advent of hardware and availability of on-the-shelf software, future deep space flight missions should consider development of a joint high-speed AACS and CDS software simulator. This could lessen the burden of spacecraft operations on the ITL, so

that resource can be more dedicated to FSW development and validation, as it was originally intended. Consequently a functional HSS can reduce the cost of the spacecraft operation by eliminating the ITL operations and maintenance cost during the later phases of cruise and orbit.

D. Spacecraft Complexities

1. Sequence Development

The complexity of the Cassini spacecraft presented multiple challenges for the sequence development process. With the DO team, the distance to Saturn, and the complex operation of the spacecraft, simply planning the activities a short time in advance like other Earth or Mars missions was not an option. To perform the constraint checks and activity coordination, observations were developed in sequences that began development up to 10 years in advance of execution. Multiple sequences were developed in different phases at a given time. For instance, a sequence planned for the 2008 time frame would be in implementation phase, while the sequence that was scheduled for uplink in a few weeks would be in the final phases of the SSUP process. This placed strain on both the DO team and the science planning team as team members found they were working on multiple sequences simultaneously. Since increased funding to hire additional personnel for that time period was not available, most teams found ways to automate their processes to ease sequence development. The Cassini mission may be faced with flight team attrition due to the length of the prime mission. However, more automated processes and standardized procedures will help offset that potential problem.

2. Memory Management

Memory management was another challenge for the Cassini–Huygens mission. While the spacecraft had more memory capability than other missions at that time, the complexity of the commands expanded to utilize the full capacity. Because the sequences were of such great length and the instrument commands so complex, the IEB strategy was developed. Subroutines are sent to the instrument for storage in instrument memory in advance of sequence execution. During the sequence execution, instruments will include commands to load these IEB subroutines when needed. This reduces the amount of memory needed for each sequence by loading these commands into the instrument memory rather than including them as part of the sequence commands.

In addition, some instruments developed cyclic commands to minimize sequence size. These commands are a standard set of activity definitions that are included at the beginning of the sequence. Throughout sequence execution, the instrument will refer to that cyclic definition through a single command rather than repeat the entire set of commands for that activity.

The sequence memory is divided into regions for ease of management during execution. These regions are specified for program types such as movable blocks, inertial vector updates, orbital trim maneuvers, minisequences, and the background sequence components. Coordination between sequences is another memory management challenge. To accommodate both the current and upcoming

sequences in the available background sequence memory region, each sequence is divided into three programs: current master, overlap master, and long-term master. During the sequence development process, the current master is set to expire at the time that the next sequence will begin loading on the spacecraft. The overlap master and the long-term master contain the commands needed for the last few days of sequence execution before the next sequence program takes over. This allows the current sequence to relinquish memory for the upcoming sequence as a function of time.

V. Conclusion

The sequence development process for the Cassini–Huygens mission to Saturn and Titan is a complex process that involves years of planning and coordination across many teams. This process had to respond to challenges related to the DO structure of the core development team, through its software, web-based tools, and configuration management. In addition, the pointing strategy during spacecraft operations was complicated by the descoping efforts during the spacecraft design. Methods were developed to manage safe and effective handoff of the spacecraft orientation between instrument teams. Because the Cassini spacecraft is so complex and has a vast range of science objectives to satisfy, the science and engineering activities are planned years in advance. To take advantage of the latest ephemeris knowledge, the Cassini operations team utilized pointing margin, timing flexibility, and update processes to allow for incorporation of this information at later phases of sequence development and execution. In addition, the extensive flight rules needed for this mission added difficulty to software development and simulation. To accommodate the complexity of the science and engineering sequences, the sequence memory must be managed carefully by preloading instrument-specific commands before the sequence, using cyclic definitions for repeat commanding, and dividing the background sequence into sections that expire when more memory is needed for upcoming sequences. The lessons learned as a result of the challenges encountered during the Cassini–Huygens sequence development process will serve as a model for future complex mission operations.

References

[1] Paczkowski, B., Gross, J., and Magee, K., “Science Planning Design Workshop,” *Proceedings of the Science Planning Process Design Workshop*, Jet Propulsion Lab., Document N94-23850, Pasadena, CA, March 2001.

[2] Morales, L., “Science and Sequence Update Process (SSUP) Operations Plan,” Jet Propulsion Lab., JPL D-25961, Pasadena, CA, March 2003.

[3] Benson, R. D., and Kahn, P. B, “Distributed Science Operations for JPL Planetary Missions,” *Proceedings of the Second International Symposium on Ground Data Systems for Space Mission Operations*, Pasadena, CA, 1992.

[4] Wessen, R. R., and Finnerty, D. F., “Science Planning and Sequencing for Cassini,” Jet Propulsion Lab., Document N94-23871, Pasadena, CA.

IX. Operations Experiences

This page intentionally left blank

Chapter 34

Rosetta Planning Concept for Pre-Comet Scenarios Illustrated with Deep Impact Observations

Kristin R. Wirth,* Viney Dhiri,† Ruben Solaz,‡ and Detlef Koschny§

European Space Agency, Noordwijk, The Netherlands

I. Introduction

THE International Rosetta mission managed by the ESA was launched on 2 March 2004 to rendezvous with comet 67P/Churyumov–Gerasimenko (C–G) in May 2014. After having placed a lander on the comet's surface, the Rosetta orbiter will continue to orbit C–G and accompany the comet through perihelion. The science mission will last about 18 months. The spacecraft carries 12 experiments on the orbiter plus 10 experiments on the lander, which will image the comet, measure its chemical and physical composition, and study its magnetic and electrical properties. For the first time the development of the cometary activity and the processes in the surface layer and inner coma are observed from a very close distance [1, 2, 3].

The Rosetta spacecraft systems and payloads were successfully commissioned from March to October 2004. In March 2005 Rosetta performed the first Earth swingby. Rosetta will make use of two more swingbys at the Earth and one Mars swingby to reach C–G. Regular payload checkouts (two or three per year) are conducted until Rosetta enters into Deep Space Hibernation Mode in July 2011. Rosetta will also perform close flybys at two asteroids, namely 2867 Steins in September 2008 and 21 Lutetia in July 2010. The global characterization of these flyby asteroids, their dynamics, surfaces and environments, and their relationship with comets form part of the main mission science objectives.

---

*Deputy Rosetta Science Operations Manager, ESTEC, Research and Scientific Support Department.

†Science Operations Engineer, ESTEC, Research and Scientific Support Department.

‡Spanish Trainee, ESTEC, Research and Scientific Support Department.

§Rosetta Science Operations Manager.

In addition, Rosetta observed the encounter of the NASA Deep Impact (DI) probe with comet 9P/Tempel 1. This scenario constituted the first active science phase of the Rosetta mission. The data from the remote sensing instruments onboard Rosetta have contributed to measure the composition of the crater and its ejected material, and to determine the changes in natural outgassing produced by the impact.

Rosetta was in a privileged position for its remote sensing instruments to observe the event. The distance between Tempel 1 and Rosetta was about 80 million kilometers, which compares to 130 million kilometers as seen from the Earth. The comet was viewed from Rosetta at an angle of 90 deg from the sun, whereas the angle from the Earth was 104 deg. Rosetta monitored Tempel 1 continuously over an extended period from 5 days before the impactor hit the comet on 4 July 2005 to 10 days afterward. Thus Rosetta observed three comet rotation periods before the impact and completed only after the activity triggered by the impact had decreased again. The main advantages of Rosetta as an observation platform compared with ground-based telescopes were the absence of a day-and-night cycle and an absorbing atmosphere that made uninterrupted and stable measurements possible.

The OSIRIS imaging system used its narrow angle camera and wide angle camera to take images of the dust and gas of the coma with different filters. Very accurate photometry of the unresolved nucleus with complete time coverage was performed. The evolution and composition of the impact cloud were monitored. In particular, the water production by the impact and the dust/ice ratio were determined [4, 5, 6]. Also, because of the different observing angles from Rosetta and Earth, three-dimensional stereo reconstructions of the coma are possible.

The ultraviolet spectrometer ALICE detected strong lines of neutral hydrogen and oxygen atoms throughout the observation period and weak lines of neutral carbon atoms on some occasions. Except for a possible enhancement in carbon emission, no changes were found in the ultraviolet spectra as a result of the impact [7].

The microwave spectrometer MIRO measured the water production rate, which did not increase significantly in the post-impact phase. The water production rate was determined to be less than had been anticipated based on models.

This chapter presents the planning concept that has been developed to coordinate the payload activities during the cruise to C–G. The Deep Impact observations scenario is used as an example to illustrate the planning strategy. The next section gives an overview of the operations executed during the DI scenario, i.e., the result of the planning process. Some typical characteristics and constraints that must be taken into account in the planning of all payload operations are addressed here. Following this, all of the planning steps required to plan the scenarios before arrival at the target comet are discussed in detail. Then the available planning tools are briefly described. Finally, the lessons learned and an outlook of the ongoing developments for the continuing mission planning are given.

II. Deep Impact Observations Scenario

Figure 1 represents the operations of the Deep Impact observations scenario that were executed onboard the Rosetta spacecraft. The experiment operations and the pointing profile are closely linked.

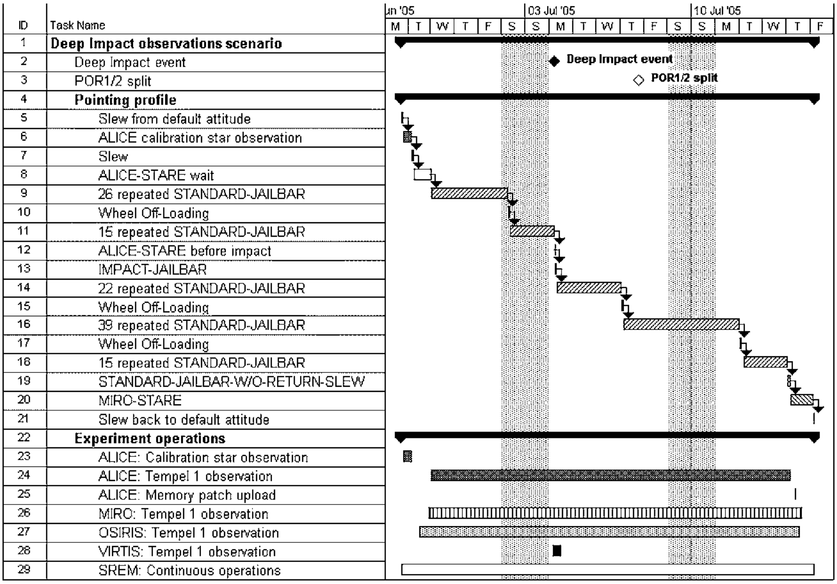


Fig. 1 Timeline chart of the Deep Impact observations scenario. The pointing profile consists of several pointing modes, slews, and wheel off-loadings (WOL). Experiment activities are shown below the pointing profile.

OSIRIS, ALICE, and MIRO monitored Tempel 1 continuously throughout the full available time period. In addition, the ALICE observations of the comet were preceded by a calibration star observation and followed by a memory update. VIRTIS, the visible and infrared mapping spectrometer, observed the comet only for a few hours around the impact, but the outburst was not energetic enough to reach the minimum sensitivity level required. The Standard Radiation Environment Monitor (SREM) continuously operates during the cruise phase to C–G to collect science data and forms part of the onboard monitoring system.

The pointing profile of the DI scenario started on 27 June 2005 with the slew from the default attitude [gyro-stellar ephemerides phase (GSEP)] to the ALICE calibration star. On the next day Rosetta slewed to Tempel 1. The comet was tracked until 15 July 2005, when the slew back to the default attitude was performed. The high-gain antenna could be pointed to Earth at the same time so that the observations could continue during the data downlink.

The detailed pointing profile was complex, because ALICE and MIRO have small fields of view (FoV) that do not overlap because of boresight offsets, i.e., the two instruments cannot observe a point source (like Tempel 1 from 8×10^7 km distance) at the same time. The ALICE FoV is a narrow slit, and to obtain spatial information in the direction perpendicular to the slit, a raster of five stopped positions was requested. Raster patterns perpendicular to the ALICE slit constitute a frequent pointing mode, and the term "jailbar" is commonly used by the Rosetta teams to describe this mode. The last "jailbar" point was compatible with the

small circular FoV of MIRO. The standard “jailbar” sequence was repeated almost continuously while Tempel 1 was tracked, except for the impact “jailbar” sequence with shortened dwell times for increased temporal resolution.

The “jailbar” repetitions were interrupted every five days for reaction wheel off-loadings (WOL). During the WOLs, the OSIRIS doors were closed and ALICE was switched off, so that contamination and high-voltage discharge by the thruster exhausts were avoided.

The duration of the standard “jailbar” sequence was 3 h, i.e., 8 standard “jailbar” sequences fit into a day, so that the pointing profile was repeated each day at the same times (apart from the discontinuity at the impact). WOL windows of 3 h were allocated, although the actual wheel off-loadings were performed in significantly less time. The use of these repetitive blocks greatly simplified the planning and allowed faster replanning and rejoining of the original timeline when a minor anomaly occurred. Furthermore, the WOL windows were used as maintenance slots, because the payload activity was very low in these slots.

The level of interactivity during the DI observations was low. As the predictions of the brightness increase of Tempel 1 caused by the impact were uncertain, the experiment teams were given the possibility to change measurement parameters in the Payload Operations Request (POR) files waiting for uplink. The execution of the updated sequences started about 3.5 days after the impact. This time is indicated in Fig. 1 by the milestone labeled “POR1/2 split.”

III. Planning Schedule

Several groups participate in the planning process. The system architecture of the Rosetta ground segment is shown in Fig. 2. The principal investigator (PI) teams have built the instruments and are responsible for the operations planning

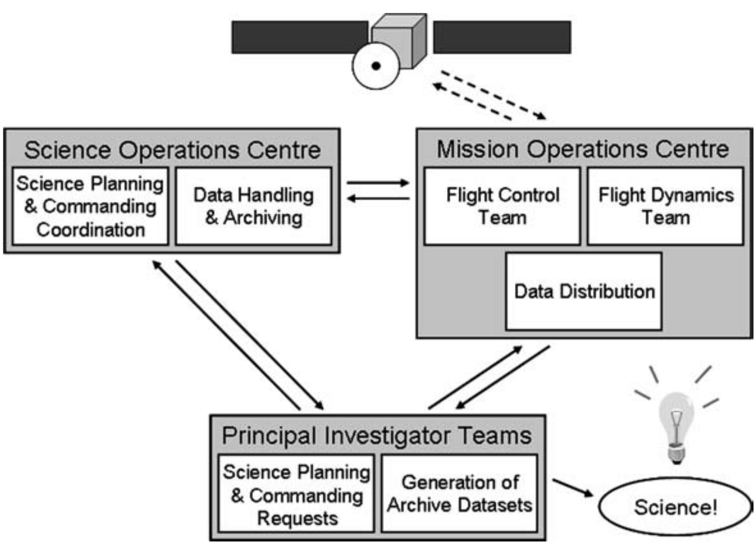


Fig. 2 System architecture of the Rosetta ground segment.

for their individual experiments. The PI teams are located at many research institutes and universities in Europe and the United States. The Rosetta Mission Operations Centre (RMOC) is responsible for the operations planning for the spacecraft platform systems and for actually "flying" the spacecraft. RMOC mainly consists of the Flight Control Team (FCT) and the Flight Dynamics Team (FDT). RMOC is located at ESA's European Space Operations Centre (ESA-ESOC) in Darmstadt, Germany. The Rosetta Science Operations Center (RSOC) is responsible for the coordination of the operations of the science payload. RSOC is located at ESA-ESTEC in Noordwijk, the Netherlands. In addition to operations planning and commanding of the payload and spacecraft (i.e., the "uplink" part), the Rosetta ground segment is also responsible for handling and archiving of the data returned from the spacecraft (i.e., the "downlink" part). However, data handling and archiving is not discussed in this chapter.

The planning concept defines the interfaces and schedules for the planning process involving the groups described previously. The following steps are identified: iteration on the top-level requests for payload and spacecraft operations, iteration on the detailed operations with emphasis on pointing, iteration on the operations request files, lead time to execution, execution on the spacecraft, and reporting. These steps are described in detail in the subsequent sections. Figure 3 shows an overview of the interactive planning process.

A. Iteration on Top-Level Requirements

Figure 4 gives the details of the first planning step, i.e., the iteration on the top-level requests for payload and spacecraft operations. The planning process is

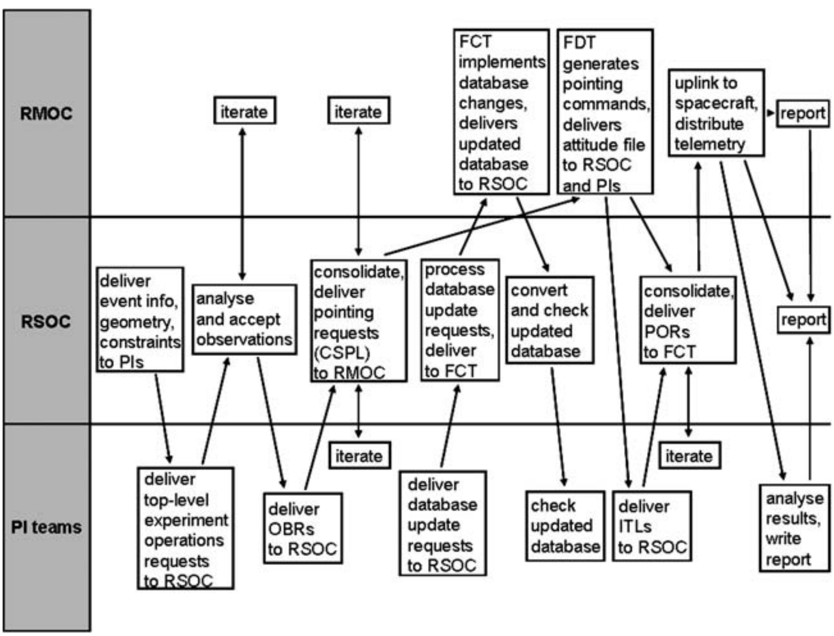


Fig. 3 Flowchart of the planning process.

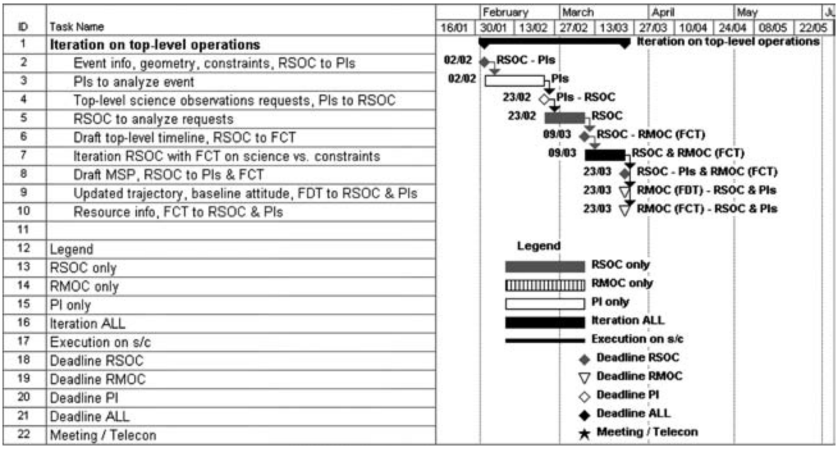


Fig. 4 Iteration on top-level operations requests.

kicked off by an official announcement of the operational scenario from RSOC to the PI teams. This announcement describes the characteristics and objectives of the scenario. It gives an overview of the geometry and states the already known operational constraints, e.g., attitude and thermal constraints, required spacecraft operations, availability of passes, and propagation delay. It also includes the planning schedule and the planning constraints, e.g., extent of interactivity, and possibility of parallel experiment operations. RSOC and the FCT work together on the preparation of the announcement.

After the announcement has been made, the PI teams work out their top-level experiment operations requests and submit them to RSOC. For each observation, the following key details are provided in an informal e-mail or document: description and objectives of the observation, pointing requirements, duration, rough estimate of the required power and data volume, reference to procedures or sequences, interactivity or visibility requirements, and special environment or spacecraft constraints.

Subsequently, RSOC assesses the feasibility and priority of the requested observations and proposes a draft top-level timeline. The FCT and RSOC iterate on the attitude and spacecraft operations. The PI teams may be asked for clarifications if necessary. At the end of this process, RSOC issues the first version of the Master Science Plan (MSP). The MSP is the main document of an operational scenario describing the detailed planning of the payload operations. It is a “living” document as it is updated several times during the planning process at well-defined milestones.

At the same time as RSOC distributes the MSP, the FDT provides an updated trajectory file and an attitude file based on the spacecraft operations (payload pointing requests are not yet detailed enough). The FCT delivers detailed resource information, i.e., available power and data rate profiles. The information from the FCT and FDT is needed in the next planning step.

B. Iteration on Detailed Operations with Emphasis on Pointing

Figure 5 gives the details of the second planning step, i.e., the iteration on the detailed operations with emphasis on the pointing profile. In addition, all database update requests must reach the FCT at the end of this step.

In the beginning, a short iteration on the top-level experiment operations requests as documented in the MSP is performed by the PI teams and RSOC to prepare the submission of the formal Observation Requests (OBR) from the PI teams to RSOC. The OBR is a filled-in form sheet containing detailed information about the requested observations with focus on the pointing requirements. It includes target details, a very detailed description of the requested pointing (e.g., a raster or scan), proposed pointing events, an improved estimate of the required power and data volume, science and special operational constraints, and an operations overview.

Based on the OBRs, the detailed pointing profile is worked out with RSOC leading, and the PI teams and RMOC deeply involved. Weekly teleconferences are held. Pointing requests from different experiments are combined. For example, in the DI scenario, the distance of the ALICE jailbar points was adjusted so that MIRO could be accommodated in the last jailbar point, and then the dwell times on the five jailbar points were negotiated. Slots are assigned to incompatible pointing requests, taking into account science priorities and operational constraints. A rough resource analysis is performed, e.g., the estimated data volume generated per day is compared with the data volume that can be downlinked per pass. Toward the end of the process, a coordination meeting between RSOC, the FCT, and the FDT takes place to finalize the pointing profile. RSOC produces the Consolidated Scenario Parameter List (CSPL) and updates the MSP. The CSPL defines the pointing modes, and the timeline for execution of these pointing modes is given in an attachment to the MSP. Operational constraints are taken over from the OBRs to the MSP. The PI teams, RMOC, and RSOC must agree on the final CSPL and new MSP version.

When the pointing profile is planned, the most intense iterations of the whole planning process take place, because the requirements of different experiments and the spacecraft are closely linked. In the schedule for planning the DI scenario, four weeks were allocated for the iteration of the pointing profile. This

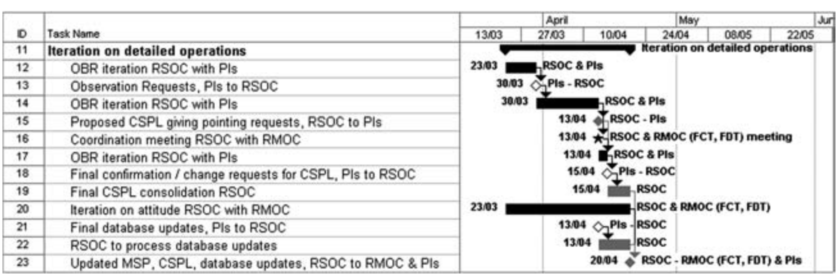


Fig. 5 Iteration on detailed operations, pointing profile, and database updates. The legend is given in Fig. 4.

turned out to be too tight, so that the pointing plan was completed with a delay of one week (which was recovered later). Therefore, two weeks are added in the planning schedules for all future pre-comet scenarios. It should also be noted that inputs from RMOC are very important from the beginning of the planning of the pointing profile.

As a preparation of the next planning step, all database update requests must be submitted from RSOC to RMOC together with the final CSPL and updated MSP. Before, the database update requests are submitted from the PI teams to RSOC and processed at RSOC. RSOC implements the updated command sequences in the RSOC planning database, so that operational request files can be analyzed without waiting for the FCT to deliver the new operational database.

C. Iteration on Operations Request Files

Figure 6 gives the details of the third planning step, i.e., the iteration on the operations request files. In the first part of this step, several activities are going on in parallel. The FDT generates the commands to execute the pointing profile on the spacecraft, and produces the final trajectory and attitude files. The FCT processes the database update requests and delivers a new operational database, which then is converted into the format used at RSOC and checked by RSOC. The PI teams and RSOC iterate on preliminary operations requests files in Input Timeline (ITL) format. The ITL indicates at which times which command sequences are to be run by the experiments onboard the spacecraft, and assigns values to formal parameters (if different from defaults). The ITL is an ASCII file that humans can conveniently read.

RSOC confirms that the experiment operations in the ITLs are consistent with the planned pointing profile. RSOC also verifies the syntax and database consistency of the ITLs and checks that no experiment constraints are violated. In addition, RSOC performs an analysis of the resources, i.e., the power consumption of the experiments, and the data volume generation and downlink.

Once the Flight Dynamics products are delivered and the database has been propagated to all systems, the PI teams and RSOC perform the final ITL iteration. After the PI teams have officially submitted their experiment ITLs to RSOC, RSOC converts the ITLs into one or several POR files. The POR contains the

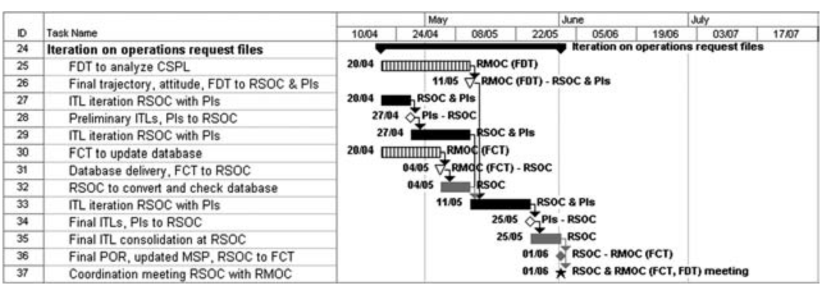


Fig. 6 Iteration on operations request files. The legend is given in Fig. 4.

same information as the ITL, and it is also an ASCII file, but the format is difficult to read for humans. The POR is the format accepted by the FCT. After POR generation from the ITLs, RSOC submits the PORs to the FCT.

Around the time of the final POR delivery, another meeting between RSOC, the FCT, and the FDT is held to make sure that all teams have the same understanding of the planned operations, to clarify all open questions, and to coordinate the subsequent team activities.

D. Execution on Spacecraft

Figure 7 gives the details of the planning steps after the POR delivery, i.e., the lead time to execution and the execution on the spacecraft. The FCT requires a lead time of at least three weeks between the reception of the POR from RSOC and the uplink of the first commands to the spacecraft. During this time, the FCT performs checks, merges the operational request files from RSOC and the FDT, adds the spacecraft operations, and works out the uplink strategy. The Mission Timeline (MTL) onboard the spacecraft can only hold 3000 commands, which is sufficient for several days of operations, but usually not for a whole scenario, and so the commands have to be uplinked in several blocks.

About one week before the start of the uplink, the PI teams make a GO–NOGO decision. GO means that the experiment operations are executed as planned; NOGO means that the participation of the experiment in the scenario is canceled completely. In case of a NOGO from one or more experiments, RSOC generates new PORs with all command sequences of the withdrawn experiments removed. A NOGO is only given in rare cases when experiment issues cannot be resolved within the expected time. For the DI scenario, all experiments were GO.

The first block of commands is uplinked to the spacecraft at least two passes before the start of the operations, in case that a pass is missed, e.g., due to problems with the ground station.

In the DI scenario, the PI teams were given the possibility to change measurement parameters after the impact. Therefore, the experiment operations were split into two parts with separate PORs. Both POR1 and POR2 together were delivered

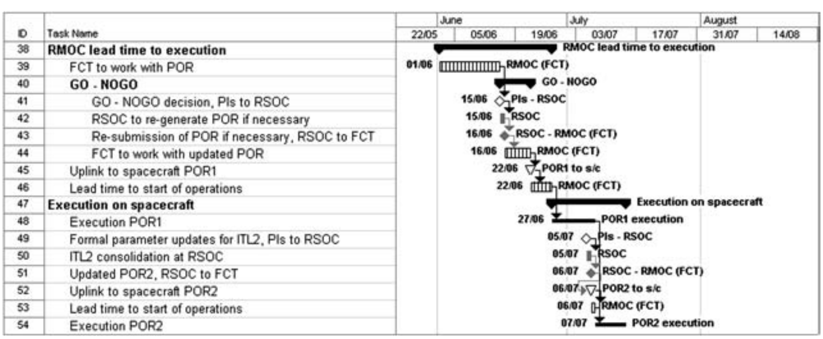


Fig. 7 Lead time, execution on spacecraft. The legend is given in Fig. 4.

to the FCT for checking and preparation of the uplink strategy, but POR2 was kept on ground. The deep impact happened just before the start of a pass, so that the first post-impact data were received with only a short delay. Based on a quick analysis of these post-impact data, the PI teams were allowed to update formal parameters in the ITLs for the second part of the experiment operations. The deadline for the delivery of the updated ITL2s from the PI teams to RSOC was 36 h after the impact, and the deadline for the delivery of the updated POR2 from RSOC to the FCT was another 12 h later. Thus, the FCT received the updated POR2 around the beginning of the third pass after the impact, and they processed and uplinked it during the same pass. The next pass was reserved as backup, and shortly afterwards the execution of the updated POR2 started onboard the spacecraft, which corresponded to about 3.5 days after the impact.

E. Reporting

The results and any problems of each operational scenario are documented for future reference and as a starting point to follow up all anomalies encountered. The FCT writes the mission operations report focusing on operational issues of the spacecraft platform and payload. RSOC writes the payload operations report consolidating inputs from the PI teams and the FCT. The results of the observations are compared against the objectives. The RSOC report also includes issues that were encountered during the payload operations, but that do not show up as anomaly reports or out-of-limits, i.e., that are not included in the RMOC report. Furthermore, feedback on the planning process is collected to identify areas of improvement. An important part of the RSOC report is the resource analysis where the actual power and data volume requirements are compared against the simulations. In this way the modeling of the experiments by the planning tools at RSOC is improved.

IV. Planning Tools

A. Experiment Planning System

The Experiment Planning System (EPS) is employed to consolidate operations request files from the experiments. The EPS characteristics and functionalities are summarized as follows:

1) The EPS uses a copy of the database defining all commands and sequences.

2) The EPS uses models of the experiments stored in experiment description files (EDF).

3) The EPS checks the syntax and database consistency of operations request files.

4) If the EPS finds a violation of a constraint defined in the EDFs, it generates a conflict.

5) The EPS simulates the power consumption of the experiments, and the data volume generation and downlink, and it produces output tables that list the resource requirements vs time.

6) The EPS produces PORs from ITLs.

7) The EPS allows to schedule sequences relative to events. Events in ITLs must be agreed by RSOC and the PI teams, and events in PORs must be agreed between

RMOC and RSOC. Events in the ITLs can be resolved to absolute times when generating the POR, or the events in the ITLs can be resolved to other events in the PORs, or the events are taken over from ITL to POR without any resolution.

In the DI scenario, the master event was the Deep Impact event, which corresponded to the impact time as seen from Rosetta [4 July 2005, 05:49 UTC (universal time coordinated) actual]. In addition, RSOC defined auxiliary events with a fixed offset relative to the DI event to facilitate synchronization of the experiment operations with the “jailbar” points. The PI teams scheduled all of their sequences in the ITLs relative to these auxiliary events. The ITL iteration took place in April and May 2005 when the range of adjustment of the impact time was 1 h. By the end of May, the prediction was accurate to +/− 3 min (3-sigma). All events were resolved to absolute times when the PORs were produced from the ITLs. The accuracy of the prediction was sufficient for the scheduling of the experiment operations, and a further improvement in the precision was only expected a couple of days before the impact, when the observations onboard Rosetta were already running. In addition, the experiment operations depended on the pointing profile, which could not be scheduled against events.

Figure 8 shows the science data volume generation and dumping vs time during the execution of the DI scenario as modeled by the EPS. Each experiment is assigned its own partition on the solid-state mass memory (SSMM) onboard the spacecraft. Data accumulate in the experiment partitions on the SSMM out of pass and are downlinked to Earth in pass. With daily passes this basically results in a sawtooth profile. On most days, all data on the SSMM can be downlinked during the pass, so that the SSMM is empty at the beginning of the next non-coverage period. However, on the impact day (day 7 in Fig. 8), a much higher data volume than on the other days is produced, mainly by OSIRIS and VIRTIS. After that, five passes (until day 12) are needed to clear the backlog.

Note that on days 7 and 8 the dumped data rate of MIRO is much lower than the data rate of OSIRIS and VIRTIS. Only on day 9, when the SSMM partitions of both OSIRIS and VIRTIS are emptied, the MIRO data rate increases again. This is caused by the different data packet sizes of the experiments and the round-robin system that dumps one packet from each experiment (if available) and repeats in

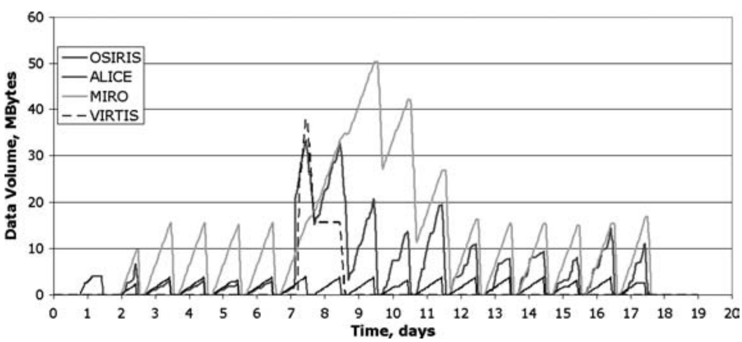


Fig. 8 Data volume analysis. Predicted science data volume generation and dumping vs time.

a loop. MIRO has very small data packets, and therefore is at a disadvantage when dumping in parallel with other experiments.

Figure 8 represents an EPS simulation under the assumption of equal dump priorities of all experiments. In reality, MIRO was given dump priority on the day after the impact, because the MIRO data from shortly after the impact were needed to decide on the update of the measurement parameters for the second part of the operations.

B. Project Test Bed

The Project Test Bed (PTB) is an environmental simulator. It models the geometry of the spacecraft with respect to the sun, planets, moons, and target comets and asteroids. It uses a model of the spacecraft, and it reads in the trajectory and attitude of the spacecraft (either from orbit and attitude files delivered by the FDT, or from pointing request files, or from a built-in orbit propagator), and the orbits and orientations of the solar system bodies. The PTB generates a three-dimensional visualization of the spacecraft and its environment, and shows, e.g., the field of view of a camera. It produces output tables with geometry parameters, e.g., position and velocity coordinates, solar elongation, phase angles, latitude and longitude of a target site on a planet, and spacecraft and solar zenith angles. It can also generate events, e.g., the spacecraft approaches a landmark at a certain distance.

In the planning process of the DI scenario, it was used to generate tables of geometry parameters, i.e., the position of Tempel 1, sun, and Earth as seen from Rosetta. Furthermore, the final attitude file delivered by the FDT that was based on the commands for the Attitude and Orbit Control System to be executed on the spacecraft, was double-checked before uplink to rule out any misunderstandings possible with a human pointing planning interface.

V. Lessons Learned and Improvements

Although the planning process for the DI scenario mostly went smoothly, important lessons have been learned for both pre-comet scenarios and the science operations at C–G:

1) Repetitive blocks of operations and maintenance slots greatly simplify the planning and resolution of anomalies, in particular when they repeat on an hourly or daily basis.

2) RMOC should be involved in the iteration of the pointing profile from the beginning to avoid “last minute” changes to an already mature pointing profile.

3) The time allocated for the pointing iteration initially was too short. Six weeks rather than four weeks should be foreseen.

The planning process for the DI scenario was supported by the EPS for manipulating operations request files, analyzing resources, and checking constraints. The PTB was used for analyzing the observation geometry. The planning tools required for continuous intense science operations are not yet existing because the start of science production corresponding to the mission science objectives was only foreseen several years from now. Going through the planning process with the available tools involved a lot of manual activities and a heavy workload for all

participating teams. The following improvements have already been made or are foreseen for the near future:

1) Since the DI scenario, the modeling of the experiments and data downlink in the EPS was improved considerably.

2) Mapping and Planning Payload Science (MAPPS), a tool complementary to the PTB, has been introduced for visualization of timelines and geometrical parameters.

3) An automated pointing planning interface is being designed.

4) A procedure and tools will be developed for implementing short-term changes of operations requests in case of anomalies, e.g., how to delete incorrect commands that have already been uplinked to the spacecraft.

VI. Conclusion

The Deep Impact observations scenario and all the other scenarios executed during the Rosetta cruise to 67P/Churyumov-Gerasimenko have some characteristics in common that clearly distinguish them from the main mission at the target comet. The total duration of all payload operations of the DI scenario was about 17 days. This is typical for the pre-comet scenarios, which have a limited duration between several days and several weeks. The planning of the DI activities onboard Rosetta started about five months before the start of the observations. Longer planning cycles for cruise scenarios of up to one year are acceptable, especially if more than one scenario is planned in parallel. After the impactor hit comet Tempel 1, the experiment teams were allowed to update measurement parameters. This low level of interactivity is common for all pre-comet scenarios except Active Payload Checkouts (which are more similar to commissioning). Results obtained during the execution usually do not feed back to updates of the remaining operations of the ongoing scenario, apart from very minor changes.

The planning concept defines the interfaces and schedules for the planning process involving the experiment teams, the Flight Control Team, the Flight Dynamics Team, and the Rosetta Science Operations Centre. The following planning steps are identified: iteration on the top-level requirements for payload and spacecraft operations, iteration on the detailed operations with emphasis on pointing, iteration on the instrument timelines, lead time to execution, execution on the spacecraft, and reporting. The planning process is supported by software tools for manipulating command files, analyzing resources, checking constraints, and analyzing the observation geometry.

In summary, the Deep Impact observations scenario was planned and executed very successfully. It was the first important active science phase for the Rosetta mission and constituted a major operational test involving complex and long-duration science operations. Valuable experience was gained that will be used to design the planning concept and operations in the comet phase.

Acknowledgments

We thank Jorge Diaz del Rio, Alejandro Cardesin, Joe Zender, and Gerhard Schwehm for their expert support of this work. The outstanding contribution and dedication of the Flight Control Team, Flight Dynamics, and the PI teams is

acknowledged and highly appreciated. We also would like to thank the TEC-SWM group at ESA-ESTEC who provide the planning tools.

References

[1] Schwehm, G., and Schulz, R., "The International Rosetta Mission," *Laboratory Astrophysics and Space Research, Astrophysics and Space Science Library*, Vol. 236, Kluwer, Dordrecht, The Netherlands, 1998, p. 537.

[2] Schwehm, G., and Schulz, R., "Rosetta Goes to Comet Wirtanen," *Space Science Reviews*, Vol. 90, No. 1/2, Oct. 1999, pp. 313–319.

[3] Glassmeier, K.-H., Boehnhardt, H., Koschny, D., Kührt, E., and Richter, I., "The Rosetta Mission: Flying Towards the Origin of the Solar System," *Space Science Reviews*, online Jan. 1, 2007, doi: 10.1007/s11214-006-9140-8.

[4] Küppers, M., Bertini, I., Fornasier, S., Gutierrez, P. J., Hviid, S. F., Jorda, L., Keller, H. U., Knollenberg, J., Koschny, D., Kramm, R., Lara, L.-M., Sierks, H., Thomas, N., Barbieri, C., Lamy, P., Rickman, H., Rodrigo, R., A'Hearn, M. F., Angrilli, F., Bailey, M. E., Barthol, P., Barucci, M. A., Bertaux, J.-L., Burns, J. A., Cremonese, G., Curdt, W., De Cecco, M., Debei, S., Fulle, M., Gliem, F., Ip, W.-H., Kührt, E., Llebaria, A., Lopez Moreno, J. J., Marzari, F., Naletto, G., Sabau, L., Sanz Andrés, A., Sivan, J. P., Tondello, G., and Wenzel, K.-P., "A Large Dust/Ice Ratio in the Nucleus of Comet 9P/Tempel 1," *Nature*, Vol. 437, No. 7061, Oct. 2005, pp. 987–990.

[5] Keller, H.-U., Küppers, M., Fornasier, S., Gutiérrez, P. J., Hviid, S. F., Jorda, L., Knollenberg, J., Lowry, S. C., Rengel, M., Bertini, I., Cremonese, G., Ip, W.-H., Koschny, D., Kramm, R., Kührt, E., Lara, L.-M., Sierks, H., Thomas, N., Barbieri, C., Lamy, P., Rickman, H., Rodrigo, R., A'Hearn, M. F., Angrilli, F., Barucci, M.-A., Bertaux, J.-L., da Deppo, V., Davidsson, B. J. R., de Cecco, M., Debei, S., Fulle, M., Gliem, F., Groussin, O., Lopez Moreno, J. J., Marzari, F., Naletto, G., Sabau, L., Sanz Andrés, A., and Wenzel, K.-P., "Observations of Comet 9P/Tempel 1 around the Deep Impact Event by the OSIRIS Cameras Onboard Rosetta," *Icarus, International Journal of the Solar System*, Vol. 187, No. 1, Mar. 2007, pp. 87–103, doi:10.1016/j.icarus.2006.09.023.

[6] Jorda, L., Lamy, P., Faury, G., Keller, H. U., Hviid, S., Küppers, M., Koschny, D., Lecacheux, J., Gutiérrez, P., and Lara, L. M., "Properties of the Dust Cloud Caused by the Deep Impact Experiment," *Icarus, International Journal of the Solar System*, Vol. 187, No. 1, Mar. 2007, pp. 208–219, doi:10.1016/j.icarus.2006.10.004.

[7] Feldman, P. D., Stern, S. A., Steffl, A. J., Parker, J. W., Slater, D. C., A'Hearn, M. F., Bertaux, J.-L., and Festou, M. C., "Ultraviolet Spectroscopy of Comet 9P/Tempel 1 with Alice/Rosetta During the Deep Impact Encounter," *Icarus, International Journal of the Solar System*, Vol. 187, No. 1, Mar. 2007, pp. 104–108, doi:10.1016/j.icarus.2006.08.016.

Chapter 35

Strategies for Enhancing Power Generation in Satellites with Large Local Time Angles

M. Pitchaimani,* T. Subramanya Ganesh,† P. Soma,‡
and S. K. Shivakumar§

Indian Space Research Organisation Telemetry Tracking and Command Network, Bangalore, India

Nomenclature

a = semimajor axis
ω_o = orbital pitch rate
μ = 398 000 km^{3}s^2

I. Introduction

INDIAN Remote Sensing (IRS) satellites are generally launched into polar sun synchronous orbits with satellite local time of 10:30 a.m. In general the pitch axis of the satellite body is aligned along the negative orbit normal, and the roll axis of the body is aligned along the direction of the orbit velocity vector. Attitude control of the first generation IRS satellites is performed by using gyro rates in the prime loop and conical Earth sensor (ES) for pitch and roll error measurements and a digital sun sensor (DSS) for yaw error measurements. ES updates the loop every 440 ms while the DSS updates the loop twice in an orbit. Conical axis of the Earth sensor is aligned along the pitch axis with a 20 deg offset toward the Earth viewing direction. Half-cone angle of the ES is about 45 deg. DSS measures the body yaw error over the poles when the sun rays are in the roll-pitch plane. The DSS has a field of view (FOV) of ±20 deg. At 10:30 a.m. local time the Earth-sun direction vector lies about 22.5 deg away from the orbital plane. Thus, the direction of the sun has a clearance of about 15 deg from the cone of ES. DSS will lose

---

*Operations Director, Indian Remote Sensing Satellites, ISRO Telemetry Tracking and Command Network.

†Mission Operations Engineer, ISRO Telemetry Tracking and Command Network.

‡Group Director, Mission Operations and Health Anlaysis Group.

§Director, ISRO Telemetry Tracking and Command Network.

sun presence signal from its field of view when the angle between the sun direction and the orbit plane exceeds 42.5 deg.

IRS satellites in general are provided with two solar arrays mounted along the positive and negative pitch axis of the satellite. While the solar panel mounted on the positive pitch axis is termed the anti-sun-side (ASS) panel, the one mounted on the negative pitch axis is called the sun-side panel. The shafts rotate about the pitch axis while the satellite is traversing from pole to pole across the equator. Each solar panel is mounted with two solar panel sun sensors (SPSS). The SPSS are tilt mounted toward the sun side by 15 deg. SPSS has a rectangular FOV of ±70 deg in the on-axis (along the orbital plane) and ±55 deg in the off-axis (across the orbital plane). Thus solar panels will not track sun when the angle between the sun direction vector and the orbital plane exceeds 70 deg.

IRS satellites carry three gyros for the measurements of body rates and attitude errors. The first generation IRS satellites have an onboard facility for compensation of gyro drift through ground commanding. A major portion of the gyro drifts is compensated through the drift rate control (DRC) data. Residual drifts are estimated and corrected by an onboard logic based on absolute sensor measurements.

IRS satellites are loaded with propellant, which caters for a period of five years of operations. When the inclination of the orbit is not maintained because of the non-availability of the onboard propellant, the orbit starts drifting from the sun, thereby increasing the local time angle (LTA). Generally four types of problems are encountered because of this:

1) The Earth sensor becomes noisy due to sun intrusion at around 9:30 a.m. local time condition. Thus the satellite attitude control is changed from dual head mode to single (ASS) head mode.

2) DSS yaw error updates stop at 9:00 a.m. local time condition or at about 50 deg LTA. The yaw axis is controlled only by gyro.

3) Solar panel power generation comes down below the minimum requirement of 50% at 8:30 a.m. local time condition.

4) At 8:00 a.m. local time condition the solar panel sun sensor loses sun presence signal (SPS) from the cross field of view. The solar panels are not guided toward the sun, eventually leading to large reduction in power generation, which may hasten the closure of the mission.

The IRS satellites are mounted with deployable sun-tracking solar panels for power generation. The first two problems can be overcome by small changes in the onboard configuration and operation strategies. This chapter addresses a solution for problems 3 and 4 to salvage the spacecraft with an innovative idea supported by mathematical rigor through modeling and analysis. Many allied issues have also been addressed through the mathematical treatment.

II. Innovative Idea

The idea looks at solving the power crisis as follows:

1) Performing a continuous sinusoidal yaw steering over every orbit so that the peak negative yaw bias is over the North Pole and the peak positive yaw bias is over the South Pole and having zero biases over the equators.

2) Selecting the exact geometric locations over poles such that the power generation over the northern and southern hemispheres is equal, thus satisfying a

probable payload over the northern hemisphere followed by capability for complete battery charging before entry into eclipse.

3) Checking that the solar array drive assembly (SADA) shaft rate requirements do not exceed the design capability while tracking the sun under yaw steering.

4) Checking for the absence of SPS from the solar panel sun sensor and introducing a roll steering strategy after taking into account all of the related issues.

Modeling and analysis ensures a flawless approach in operations and also cautions about possible deviations in the future. The aim is to develop the required mathematical model to compute at any instant 1) the angle of incidence of the sun rays on the solar panel, which is used for power estimation; 2) the cross track angle of the sun on the solar panel to check for a possible non-tracking due to absence of SPS from the solar panel sun sensor; and 3) the SADA shaft rate requirements under body biases to maintain continuous sun tracking by the panel.

The flight control software of the first generation IRS satellites does not facilitate performing autonomous yaw steering. Thus the yaw steering had to be carried out by inducing gyro drift of the yaw gyro through the DRC logic. Since the polarity of the yaw steering bias during the descending phase (north pole to south pole) was different from the polarity of the yaw steering bias during the ascending phase (south pole to north pole), the gyro drift rate commands had to be suitably programmed at the poles. Therefore, even though theoretically a sinusoidal yaw steering is desirable, due to operational constraints, a triangular yaw steering profile was chosen. The time tagging (TT) facility onboard permitted programming of the change in gyro drift values at the required points, namely the poles, with an accurate timeline on an orbit-by-orbit basis continuously throughout the rest of the mission life. This idea was innovative because it was the first time that a yaw steering strategy was being applied to any IRS satellite for the purpose of power generation improvement.

III. Mathematical Formulation

The development of mathematics requires a sun position vector in addition to spacecraft position and velocity vectors for every instant T:

T: year, month, day, h, min, s, ms
$[X_{sc}\ Y_{sc}\ Z_{sc}]$: spacecraft position vector = ***SCPOSVEC***
$[\dot{X}_{sc}\ \dot{Y}_{sc}\ \dot{Z}_{sc}]$: spacecraft velocity vector = ***SCVELVEC***
$[X_{sun}\ Y_{sun}\ Z_{sun}]$: sun position vector = ***SUNPOSVEC***

Spacecraft position and velocity information in the inertial frame at 30-s interval is taken for one full orbit as generated by the standard spacecraft ephemeris generation software. The sun position is predicted using a simple model published in the *Astronomical Almanac* [1].

A. Computation of Spacecraft-to-Sun Vector

The computation of the spacecraft-to-sun vector is as follows:

$$\boldsymbol{SC2SUNVEC} = \boldsymbol{SUNPOSVEC} - \boldsymbol{SCPOSVEC} \tag{1}$$

$$\boldsymbol{SC2SUNVEC\_UNIT} = \text{unit}\,(\boldsymbol{SC2SUNVEC}) \tag{2}$$

B. Construction of Orbit Reference Frame

Orbit reference frame is a coordinate system represented about the instantaneous position of the spacecraft with $\boldsymbol{X}_0$ toward the Earth center and $\boldsymbol{Z}_0$ toward the negative orbit normal while $\boldsymbol{Y}_0$ completes the right-handed triad. They are computed as given in the following:

$$\begin{aligned}\boldsymbol{X}_0 &= \text{unit vector along the negative spacecraft position vector}\\ &= -\boldsymbol{SCPOSVEC\_UNIT}\end{aligned} \tag{3}$$

$$\boldsymbol{Z}_0 = -\,\boldsymbol{SCPOSVEC\_UNIT} \times \boldsymbol{SCVELVEC\_UNIT} \tag{4}$$

$$\boldsymbol{Y}_0 = \boldsymbol{Z}_0 \times \boldsymbol{X}_0 \tag{5}$$

Here $\boldsymbol{X}_0$ represents the reference direction of the body yaw axis while $\boldsymbol{Y}_0$ represents the reference direction of the body roll axis. $\boldsymbol{Z}_0$ representing the reference direction of the body pitch axis completes the right-handed system.

C. Yaw Steering

It is well known that when a negative yaw bias is given to the spacecraft of IRS class over the North Pole, the tracking solar panels are taken closer toward the sun direction. The incident angle of the sun rays on the panel reduces and thus the power generation increases as $A \cos \theta$, where A is the maximum current at normal incidence. A negative yaw bias, however, is unfavorable at the South Pole because of geometry, while a positive yaw bias at the South Pole orients the panels closer to the sun. Thus a sinusoidal yaw steering is required over an orbit to increase the average power generation. Since a sinusoidal yaw steering cannot be commanded from ground, a cyclic triangular pattern that can be easily achieved onboard through ground commanding was proposed.

The DRC of the yaw gyro is commanded to a value, which creates a drift of +60 deg in 50 min or half the orbit period when the spacecraft descends from the north pole to south pole. If the initial bias at the north pole is –30 deg, the yaw bias will keep building up linearly and the bias over the south pole will be +30 deg. At the south pole the DRC is changed again with an appropriate value such that the drift is –60 deg in half an orbit period, thereby leaving a bias of –30 deg over the north pole. Thus by changing the DRC twice over poles in each orbit through time tagged commands, a cyclic triangular yaw steering is achieved.

During the simulation exercise it was found that the north precision yaw sensor (PYS) and the south PYS update points are the most appropriate locations to change the DRC of the gyro and initiate yaw steering. For this purpose the PYS update points are computed in advance by checking for a condition over the orbit where ***SC2SUNVEC*** makes an angle of $90 \pm \varepsilon$ deg with the

SCPOSVEC. This indirectly ensures that the ***SC2SUN*** lies in the roll pitch plane. By checking for the polarity of the third component of ***SCPOSVEC***, the place is understood as a north PYS update point or a south PYS update point. Since the PYS update points are determined by the sun's declination, making the PYS update points to be the starting points of the yaw steering, achieves a symmetrical power generation pattern by the solar panels for any declination angle of the sun. The symmetry in power generation is between the region from the north pole of the satellite orbit to the sun's latitude and the region between the sun's latitude and south pole of the satellite orbit. Ensuring such symmetry will be beneficial in the planning of payload operations and other activities, which require power considerations.

The yaw steering bias is computed from the north PYS update point to the south PYS update point as follows:

$$\text{Yaw bias} = -30 + \frac{60}{half\_orb\_period}\,\mathrm{d}t \tag{6}$$

where dt is the time elapsed from the north PYS update time and half_orb_period is half the orbit period in seconds. The orbit period is computed from the vis-viva equation using the state vectors as follows:

$$\boldsymbol{SCVELVEC}^2 = \mu\left[\frac{2}{\boldsymbol{SCPOSVEC}}\right] - \frac{1}{a} \tag{7}$$

$$a = \left[\frac{2}{\boldsymbol{SCPOSVEC}} - \frac{\mathbf{SCVELVEC^2}}{\mu}\right]^{-1} \tag{8}$$

$$orb\_period = \sqrt{\frac{4\pi^2 a^3}{\mu}} \tag{9}$$

Similarly the yaw steering bias during the ascent from south pole to north pole can be computed as

$$\text{Yaw bias} = 30 - \frac{60}{half\_orb\_period}\,\mathrm{d}t \tag{10}$$

where dt is the time elapsed from the south PYS update time.

D. Construction of Spacecraft Body Reference Frame

The spacecraft body reference frame ($\boldsymbol{X}_b\ \boldsymbol{Y}_b\ \boldsymbol{Z}_b$) is defined about the spacecraft center of mass such that this frame is coincident with yaw, roll, and pitch axes of the body. When attitude bias is not introduced, spacecraft body frame coincides with orbit reference frame. In the presence of attitude biases like the yaw steering bias ϕ, the body frame is computed from the knowledge of the orbit reference frame ($\boldsymbol{X}_0\ \boldsymbol{Y}_0\ \boldsymbol{Z}_0$) as

$$A = \begin{bmatrix} \cos\phi + e_1^2(1-\cos\phi) & e_1e_2(1-\cos\phi)+e_3\sin\phi & e_1e_3(1-\cos\phi)-e_2\sin\phi \\ e_1\,e_2\,(1-\cos\phi)-e_3\sin\phi & \cos\phi+e_2^2\,(1-\cos\phi) & e_2e_3(1-\cos\phi)+e_1\sin\phi \\ e_1\,e_3\,(1-\cos\phi)-e_2\sin\phi & e_2e_3(1-\cos\phi)+e_1\sin\phi & \cos\phi e\_2^3(1-\cos\phi) \end{bmatrix}$$

$$\left(\vec{\boldsymbol{X}}_b\vec{\boldsymbol{Y}}_b\vec{\boldsymbol{Z}}_b = A\,\vec{\boldsymbol{X}}_o\vec{\boldsymbol{Y}}_o\vec{\boldsymbol{Z}}_o\right) \tag{11}$$

All orbit reference frame axis vectors are referenced in the Earth-centered inertial frame. Euler angle and Euler axis method of attitude representation is suitable here as the yaw rotation takes place about the body yaw axis, which is an arbitrary axis in the Earth-centered inertial frame [2]. The spacecraft body axes vectors are now derived as vectors referenced in the Earth-centered inertial frame. This way the body axis vectors can be easily correlated with the sun vector, which is also referenced in the Earth-centered inertial frame.

E. Computation of Sun Incident Angle on the Solar Panel

The sun incident angle is defined as the angle between the normal to the solar panel and ***SC2SUNVEC***. The incident sun rays make two types of angles called along-track angle and across-track angle. The solar panels are guided by a control system, which takes error inputs from the SPSS. The SPSS measure the along-track error of the panel and provide a feedback to the control system, which uses a servo actuator to orient the panel such that the panel normal is the same plane as that of the ***SC2SUNVEC***. Since the along-track angle is kept near zero deg by the sun tracking panel, the ***SC2SUNVEC*** lies in the plane formed by the pitch axis and the panel normal. Thus when the panel is tracking the sun by keeping the along-track angle near zero deg, the sun incidence angle is the same as the cross-track angle. Hence, the knowledge of the orientation of the panel normal is required to compute the sun incident angle. The panel normal is computed in the following manner.

A coordinate system about the center of the panel ($\boldsymbol{X}_p$ $\boldsymbol{Y}_p$ $\boldsymbol{Z}_p$) for the sun side (SS) panel is formed such that

$$\boldsymbol{X}_p = -\boldsymbol{Z}_b \tag{12}$$

is the negative pitch axis of the body, $\boldsymbol{Y}_p$ lies in the panel plane but is perpendicular to $\boldsymbol{X}_p$, and $\boldsymbol{Z}_p$ is the panel normal.

Since for a tracking SS panel the ***SC2SUNVEC*** lies in the plane formed by the panel normal and the negative pitch axis of the body,

$$\overrightarrow{\boldsymbol{Y}_{\mathrm{p}}} = \overrightarrow{\boldsymbol{SC\ 2SUNVEC}}\boldsymbol{X}\,\overrightarrow{Z_b} \tag{13}$$

$$\overrightarrow{\boldsymbol{Z}_p} = -\overrightarrow{\boldsymbol{Z}_b}\boldsymbol{X}\overrightarrow{Y_p} \tag{14}$$

Since the panel normal vector also happens to be a vector referenced in the Earth-centered inertial frame, the incident angle of the sun ray can be computed as

$$\theta = \cos^{-1}(\overrightarrow{\boldsymbol{Z}_p}\ \overrightarrow{\boldsymbol{SC\ 2SUNVEC\_UNIT}}) \tag{15}$$

For a normal spacecraft without any attitude biasing, θ is constant over the orbit that is also the measure of the local time angle. In case of yaw steering, θ reflects the yaw steering with

$$\theta_{\text{pole}} = \text{LTA} - \text{yaw bias} \tag{16}$$

$$\theta_{\text{equator}} = \text{LTA} \tag{17}$$

At any other place, θ is in between and varying because of the triangular yaw steering being carried out onboard.

IV. Results and Discussion

The incident angle profile for a yaw steering of ± 30 deg was computed over an orbit for IRS-P3 and plotted. Figure 1 shows the cyclic triangular yaw steering pattern induced on the satellite platform. This yaw steering is performed such that the negative bias peak occurs at the North Pole while the positive bias peak occurs at the South Pole. The sun incident angle profile is shown in Fig. 2. It can be clearly seen that performing yaw steering on the satellite reduces the angle between the sun and the solar panel normal. The sun incident angles are minimum at the PYS update points and maximum at the sun latitude in tune with the yaw steering profile.

A. Current Generation Pattern and Energy Computation

IRS-P3 was generating 6 A from both panels when the local time was around 8:20 a.m. When this current was integrated over the sunlit region, it provided a capacity of only 8.0 Ah. However, when yaw steering with maximum amplitude

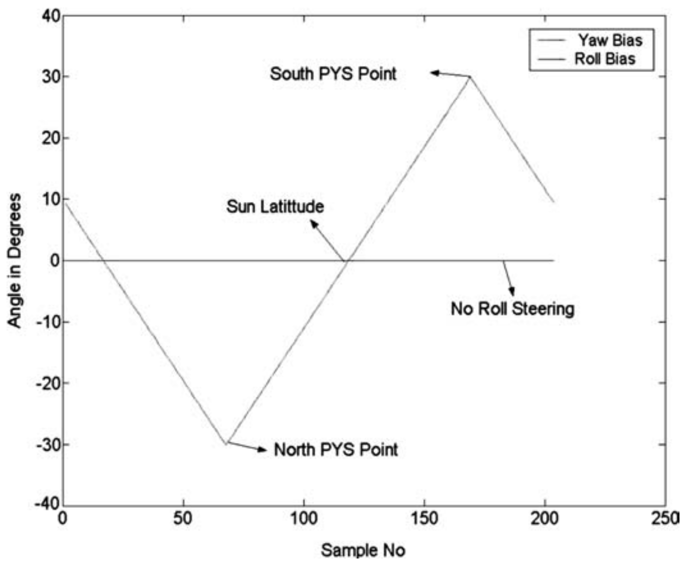


Fig. 1 Profile of triangular yaw steering over an orbit.

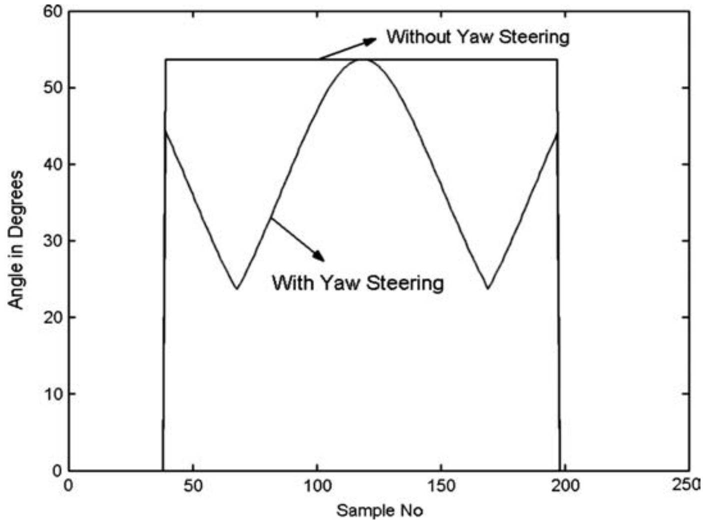


Fig. 2 Profile of sun incidence angles with and without yaw steering.

of ±30 deg was applied, a better current generation pattern was observed, which provided an integrated energy of 10.45 Ah. It is thereby seen to have an increase of 30.6%. Figure 3 explains these facts. Hence, it is found that a systematic yaw steering increases the power generation. This concept was applied to IRS-P3 from December 2004, and the spacecraft was recovered from the power crisis.

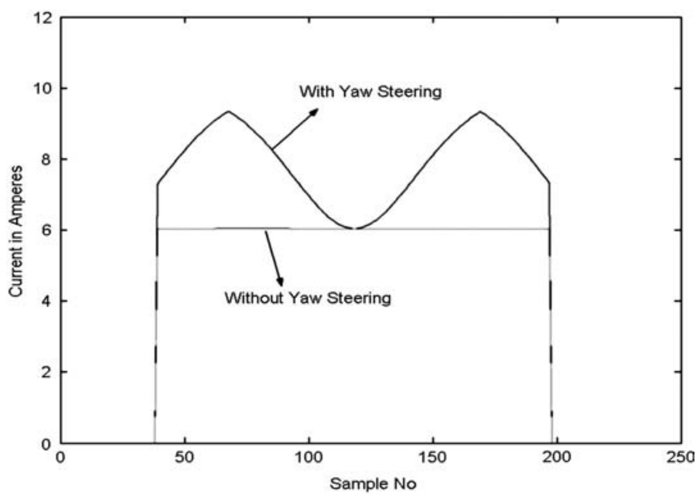


Fig. 3 Profile of solar array current generation with and without yaw steering.

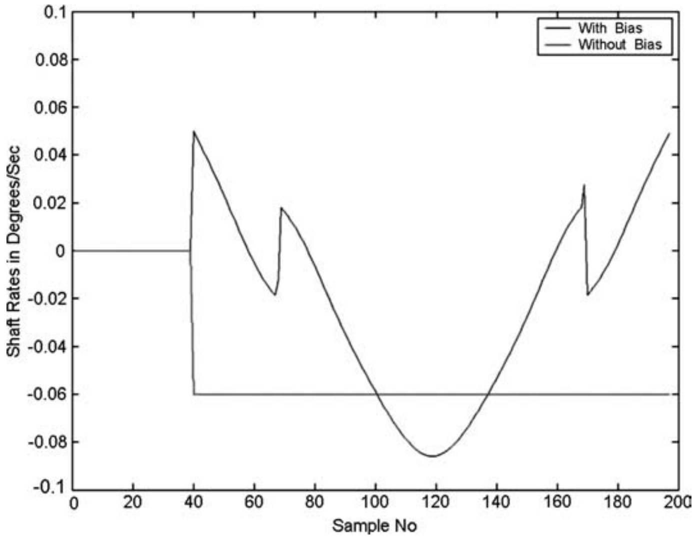


Fig. 4 Profile of solar panel shaft rotation rate requirements with and without yaw steering.

B. Varying SADA Shaft Rate Requirements

When yaw biasing is applied on the body, SADA shaft rotates at a different rate to maintain the on-axis sun tracking. In the case explained previously, the SADA shaft rotates at 0.08 deg/s at the sun latitude and at –0.05 deg/s at eclipse exit, which are above and below the nominal rate of 0.06 deg/s. However, the highest rate demanded by SADA during yaw steering is well below the maximum capability of 1.5 deg/s. Thus this does not pose any problems. Fig. 4 shows the normal rate and the rate during yaw steering.

C. Loss of SPS from SPSS and Roll Steering

As per the design specifications, the along-track field of view of the SPSS is ±70 deg and the cross-track field of view is ± 25 deg. Hence the panel may lose SPS near the equator in yaw steering mode when the LTA increases sufficiently. When that situation is encountered, it is proposed to modulate a roll steering of about 10 deg over the yaw steering in such a way that the roll steering is maximum near the sun latitude and zero near the PYS update points. This can be achieved by properly commanding the roll DRC over the equator using the TT facility. However, to perform this the spacecraft may have to be configured accordingly. Associated problems of drift and residual ω_o can be taken care of. Figure 5 shows a case of yaw and roll steering, which helps avoid loss of SPS on the SPSS besides improving power generation.

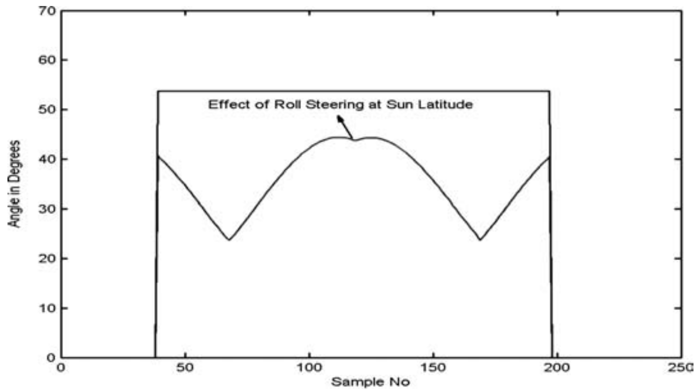


Fig. 5 Profile of solar array current generation with roll steering modulated over yaw steering.

V. Conclusion

The implementation of a cyclic yaw steering can thus help satellites with large local time angles to orient their solar panels closer to the sun direction thereby improving power generation. Modulating a roll steering over the pure yaw steering can further increase the power generation. However, the associated issues with regard to the roll steering should be addressed. Yaw steering strategy has helped in increasing the useful mission life of IRS-P3, a satellite that was launched in 1996.

Acknowledgments

The authors would like to express their gratefulness to K. V. S. S. R. Anjenayulu, Mission Director IRS-P3, for his advice and for extending his participation in operationalizing the preceding concepts. The authors would also like to express their thankfulness to J. D. Rao, Operations Director IRS-P3, and T. V. S. R. K. Prasad, Spacecraft Operations Manager IRS-P3, for their contributions in operationalizing the preceding concepts.

References

[1] *Astronomical Almanac*, Nautical Almanac Office, U.S. Naval Observatory, 1995, p. 3.

[2] Wertz, J. R., *Spacecraft Attitude Determination and Control*, Vol. 73, D. Reidel Publishing Company, The Netherlands, 2002, p. 413.

Chapter 36

RADARSAT-1 Mission Operations: Second Decade

Robert Saint-Jean*

Canadian Space Agency, Saint-Hubert
Quebec, Canada

I. Introduction

RADARSAT-1 was launched on 4 November 1995, and on 1 April 2006, the Canadian Space Agency (CSA) marked the completion of the 10th year of operations of the RADARSAT-1 system. Originally built for five years, RADARSAT-1 has now more than doubled its life expectancy with still better than nominal product quality. The system provides a large variety of spaceborne Synthetic Aperture Radar (SAR) data products in C-band (5.300 GHz) to a community of users. The imagery is used in a number of applications ranging from natural resources mapping (agriculture, geology, deforestation, etc.), environmental monitoring (global warming, ice motion, illegal oil dumping, etc.), national security (ship monitoring, illegal ship traffic, etc.), or disaster mitigation (floods, oil spills, hurricanes, etc.) using techniques such as simple image interpretation, data fusion, change detection, or interferometry applications.

RADARSAT-1 is the first operational civilian Earth observation SAR system. RADARSAT-1 is owned and operated by the Canadian Space Agency while the worldwide distribution of the data is provided by MacDonald Detwiller and Associates, Geospatial Services Inc. (MDA GSI). As of June 2006, with more than 55,000 completed orbits and with around 240,000 requests submitted, the system has maintained a delivery efficiency of better than 95%.

Several major achievements have made extensive use of the RADARSAT-1 data over the years. The Antarctic Mapping Mission, the Disaster Watch program, the Hurricane Watch program, the International Charter Space and Major Disasters, and the Modified Antarctic Mapping Mission are some of the major ones. These projects not only benefited from the SAR data but, to some extent, they also changed the way the operations are done.

---

*Head, Mission Planning, Satellite Operation.

Throughout the years, both the spacecraft and the ground system underwent several upgrades to correct some system deficiencies, to improve the overall efficiency and to adapt to new applications, clients demands, and technological evolution. All upgrades were done without affecting the productivity of the system thanks to a development system where patches are prepared and tested before implementation of the production systems. Furthermore, CSA introduced real-time database duplication to reduce downtime in the eventuality of a system problem. All communications links are also redundant and independent to minimize risks.

This chapter will focus on the improvements made to the various ground systems throughout the years. New technology and experience but also new projects helped optimize the available resources. We will give an overview of the operations of RADARSAT-1 from a ground system point of view and provide an insight on the improvements that are in part responsible to the success of the mission.

II. Interaction with Order Desks

From the beginning of the mission, the Order Desks (OD) are the privileged interface between the clients and the RADARSAT-1 system [1]. These ODs are identified by their role: the RADARSAT international (RSI) OD (currently operated by MDA GSI), receiving data requests from all commercial and foreign customers; the Alaska Satellite Facility (ASF) OD at the University of Alaska, Fairbanks, taking care of NASA's requests; the Canadian Ice Services, Environment Canada (CIS) OD, responsible for the Canadian Ice Services of Environment Canada; the Calibration and analysis system (CAS) OD, who monitors data quality, and finally, the Mission Control System (MCS) OD, who manages the various "Background Missions" and plans the emergency requests. Recently, the ESA Contingency OD was added to the system as a reciprocal operational service to ESA in the eventuality of a temporary failure of the ENVISAT Advanced Synthetic Aperture Radar (ASAR) system.

III. Data Acquisition Programs

A. Background Programs

As part of its original mandate, the RADARSAT-1 mission held several obligations to the community with regards to Earth observation. One of these obligations was to obtain a global coverage of SAR data to be obtained with a low priority (in background of normal activities) and that would serve as a global archive [2]. Since then, the Background Mission data have been used to monitor global changes, to collect data over environmentally sensitive areas of the world, and even to produce mosaics of continents.

In 1998, the Disaster Watch program was put in place to monitor ongoing catastrophic events around the world. Every day, a scan of ongoing disasters is made and if the system is idle, data are acquired and downlinked but not processed, to a data reception facility (DRF). This allows early detection of major catastrophic events.

The Hurricane Watch program started shortly after the Disaster Watch program. Hurricane Watch uses a concept similar to Disaster Watch but focuses on a more specific area over a limited period of the year.

Disaster Watch and Hurricane Watch data are acquired in anticipation of user's needs. In these programs, users are not actually requesting the data, but a team is

monitoring the situation around the world on a daily basis and is tasking the spacecraft. This advanced planning operation has proven its value on several occasions during a major crisis, such as the December 2004 Indian Ocean tsunami, and serves frequently as a baseline for disaster monitoring and change detection.

B. International Charter: Space and Major Disasters

In 2000, CSA joined European Space Agency (ESA) and Centre National d'Études Spatiale (CNES) in the International Charter Space and Major Disasters. The Charter ensures a 24-hour-a-day access to Earth observing systems to quickly provide data to mitigate natural disasters around the world. Since 2000, Indian Space Research Organisation (ISRO), Comision National de Actividades Especiales (CONAE), National Oceanic and Atmospheric Administration/United States Geological Survey (NOAA/USGS), DMC International Imaging (DMCii) and Japan Aerospace Exploration Agency (JAXA) have also joined the Charter.

The objective of this global initiative is to provide a free of charge access to space system products to authorities concerned with catastrophic events in the form of data and information products. These products are obtained from the various spaceborne sensors for situations where human life is in danger or when substantial damage on infrastructure or resources is expected. Using guidelines put together by application experts from the various space agencies, the 24-hour-a-day stand-by officer is able to efficiently task the available sensors to obtain the most relevant data in the shortest delays. The Charter is the first multi-agency, multimission, multisensor operational service available to the international community. The members participate on a voluntary basis with no exchange of funds.

The interoperability of the tasking system has proven its efficiency. As of 1 June 2006, the Charter was activated 105 times. The Charter web site (http://www.disasterscharter.org/) provides more details on its functions and operations.

C. Antarctic Mapping Mission and the Modified Antarctic Mapping Mission

The first major scientific achievement of the system was without doubt the September 1997 Antarctic Mapping Mission. The resulting images provided the first synoptic map of all the Antarctic ice streams, the first Radar-derived map of ice divides and catchment basins and the production of interferometrically derived ice velocity maps. Over 8000 image frames were collected during this first mission [3].

In September 2000, the Modified Antarctic Mapping Mission focused on interferometry coverage of Antarctica that provided the impetus toward tighter orbit control to allow interferometric data acquisition under routine operations [4]. A total of 2390 image frames were acquired during this second mission.

IV. Network Station Certification

The network of certified data reception facilities (Fig. 1) has evolved at a surprising rate over the years. From four data reception facilities available at launch (Gatineau and Prince Albert for Canada, Fairbanks and McMurdo for the United States) the network currently has a total of 33 facilities, including nine transportable facilities. Three additional DRF have showed interest in joining the

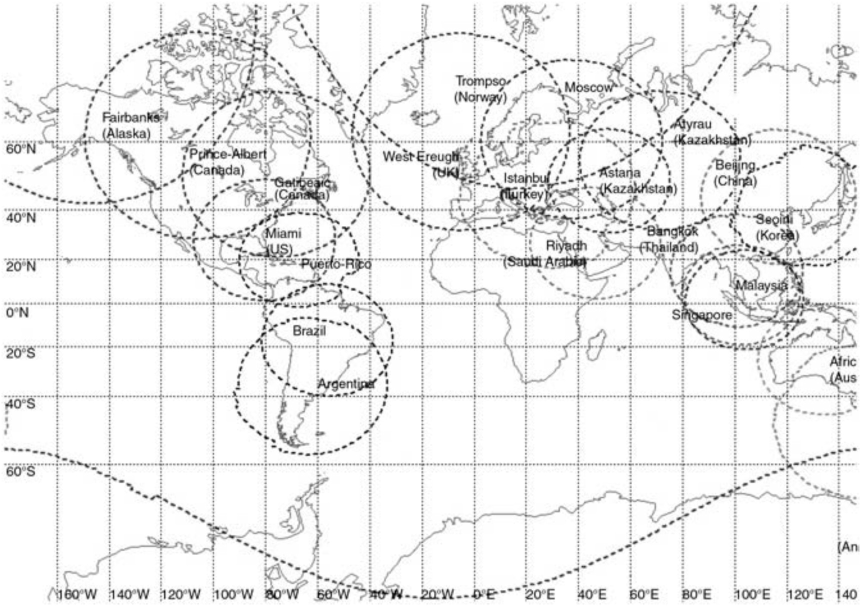


Fig. 1 RADARSAT-1 data reception facilities network as of 1 June 2006.

network in 2006. The certification process is thorough and is held jointly by CSA and MDA GSI to ensure a strict data quality control. A station certification is needed to ensure that the facility can exchange files, acquire, process, report, and archive according to the established standards. A product certification is then needed to again ensure that the facility can deliver data products to their clients that respect the published quality standards for RADARSAT-1 data.

V. Conflict Resolution, Mission Management Office Database and Request Planning Operations

A. Conflicts Resolution

Although the system has a large variety of modes and beams (Fig. 2), it can acquire only one type of data at a specific time that can lead to conflicts due to overlapping users data requests having incompatible characteristics. Other conflicts can arise due to resources availability or systems constraints. The conflict resolution process starts 14 days before first "possible" acquisition, providing enough time for discussions with clients to replan in case of conflicting requests.

MCS also has the responsibility to solve all conflicts, prepare the acquisition plan, and coordinate the transmission and reception of all operational reports with the network stations [5].

Conflicts are resolved by applying a set of guidelines that follow the original data policy [6]. The guidelines use various parameters such as priority and

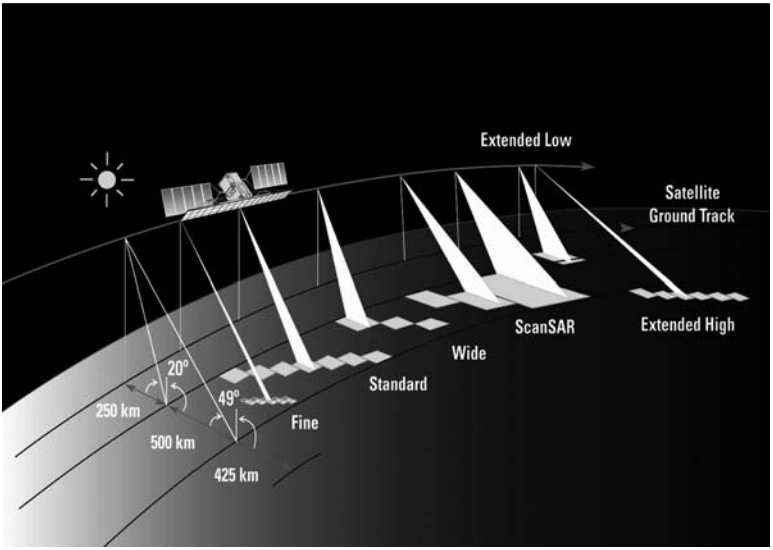


Fig. 2 RADARSAT-1 modes and beams.

submission date to solve the conflict. However, because we are seeking a higher degree of satisfaction, the concept of *favor request* was introduced. Depending on the specifics of each conflict, it is sometimes possible to replan the winning and the losing requests to create a win-win situation where conflict is resolved and clients are accommodated with minimal sacrifice of their original request. This concept is now proven useful and is used on a daily basis for the operations of RADARSAT-1.

B. Usage of Onboard Recorder Resource and Sidelays

Systems constraints have been a concern since the very early stages of the mission. Probably the most important constraint on the SAR payload system is the need for minimum image duration of 1 min. This constraint is manageable for real-time downlink to a network station but becomes quite annoying when the data need to be placed on the onboard recorder (OBR). The recorders use a relatively old magnetic tape system that shows signs of aging. Two OBR units were used until June 2002, when a first unit was deactivated following a major anomaly. Since the capacity of the OBR is limited to approximately 13 min of recorded data and the number of data downlinks has been reduced from 4 to 5 to a maximum of one per day in an attempt to extend the life of the resource, new ways to optimize the capacity of the OBR had to be found.

The sidelays mask concept was introduced as a solution to this problem and was implemented in production in early 2003. The sidelays uses a virtual mask to which data are downlinked but not recorded on the OBR or on the ground. The sidelays concept releases the constraints on the OBR system as it allows storing the OBR

only the data requested by the client while the rest of the data normally collected to satisfy the systems requirement are not recorded. Using this concept has allowed us to increase the amount of usable data stored on the OBR by up to 60%.

C. Mission Management Office System and Database

A major improvement to the system was performed early in the mission and allowed the system to be tasked at a time closer to the acquisition. Originally, the shortest possible delay between payload tasking and image acquisition was 53 h. Since 1999, this delay was shortened to 29 h [3].

The complex database system that forms the core of the Mission Management Office (MMO) has been upgraded several times due to hardware or software requirements. The planning timeline tool that is used on a daily basis by the mission planners gives a trivial example of these constraints. The system was originally built using a single alphanumeric character to represent network station masks. Some characters are reserved by the system, others for special tasks, in all, a total of 31 facilities could be supported. A complete review of the system had to be done to allow more DRF to be represented on the timeline. Currently, a two alphanumeric characters concept is in use.

The MMO was also faced with an increasing load on the databases due to accesses by the planning software, the report generation, systems queries, statistics, etc. The database systems could hardly cope with the demand. To reduce the load on the system, a concept of replication was put forward by the software support team. Planning operations are done on the production database while queries of all kinds are done on the replication database. Not only does this greatly ease and accelerate the operations, but it also it ensures a real-time backup of the core database.

Some improvements were also application driven. During the Antarctic Mapping Mission of 1997 and 2000, orbit maintenance was carefully maintained to ±1 km to allow interferometric data acquisition [4]. After the missions, the orbit was returned to its nominal orbit maintenance scenario of ±5 km. Soon after, users from around the world requested an improved orbit maintenance to allow long-term interferometric data pairs acquisition. In March 2001, CSA decided to permanently maintain the orbit within ± 2 km.

VI. File Transfer Coordination

A. Communications with Order Desks

Data requests prepared by the ODs and sent to the MMO are provided through secured web-based interface. Throughout the years, communications links have been upgraded to increase speed, security, and reliability. OD servers, originally physically located at the ODs, have now all been repatriated to CSA to improve maintainability. The ODs can access the systems through secure access (secure connection, firewall, and password protected access).

B. Communications with Data Reception Facilities

Fourteen days before the start of the execution of the 24-h long acquisition plan, conflicts are processed and a planning of the available requests takes place.

Overnight, an automated system scans the databases and prepares a Reception Request (RRQ) that is provided to DRFs to allow early planning of network station's resources. The RRQ is considered as a preliminary document because it uses a coarse orbital ephemeris. Later on, after the acquisition plan has been finalized, a new and definitive Reception Schedule (RSH) file is provided to the DRFs. The RSH is the final document and uses the best available predicted orbital ephemeris. The RRQ and RSH files are generated at specific times, and DRF are encouraged to collect their files from a secure File transfer protocol (FTP) server at regular intervals. Located at CSA, the file server went through several phases of upgrades to improve reliability (faster, more efficient servers), speed (fiber optic link), and security (secure connection, firewall protected access). Figure 3 summarizes the information exchange between the users and the various operational entities of the RADARSAT-1 system.

VII. Data Losses and Tracking of Archived Data

A. Data Losses

Once the data are acquired by the DRF, a Reception Report (RRP) is sent back to the MMO through the FTP file server to confirm or infirm the reception of the data. Upon validation of the data quality, the DRF archives the data and sends

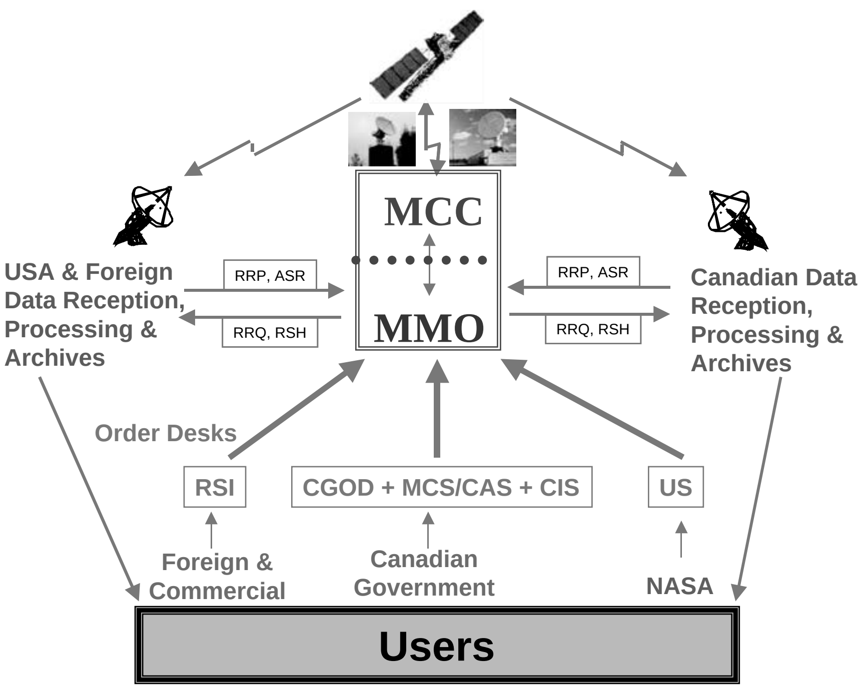


Fig. 3 RADARSAT-1 system.

back to the MMO an Archive Storage Report (ASR). All acquired data are archived locally by the facility according to the current standards or sent back to Canada for archival. If a problem is detected at reception, processing, or archival, it has to be reported as soon as possible with a Data Loss Report (DLR). The DLR is then transmitted to the MMO by e-mail and provides all necessary details to explain the situation.

Upon reception of a DLR, an investigation is started and all details are collected in the Data Loss Information System (DLIS). This database collects information on the exact timing of the reported losses and tracks all information. When the issue is understood and its extent is known, the information is published in the Image Strip Catalog (ISC), the official archive database. The ISC is accessible to all ODs so that all are aware of the availability of the data. In the ISC, data availability is flagged as total loss (data are not available), partial loss (some data have been reported as lost and the status of the rest of the data is uncertain), or no reported loss (data are available for processing). Since 22 April 2003, all new data losses are recorded in DLIS. All data losses documented before that date are being transcribed in DLIS going backward in time.

B. Archives

The RADARSAT-1 Data Policy [7] indicates that all data collected by RADARSAT-1 have to be archived and this archive has to be maintained for a period of 15 years following the end of the RADARSAT-1 mission. This presents a significant challenge considering the duration and the amount of data stored in the archives (411,482 min of recorded SAR data as of 1 June 2006). Technology has evolved at a considerable rate, and current operational requirements are much more demanding than original technology could provide. At the Canadian Archive (CARCH) facility, archives were transcribed several times to ensure the quality, the accessibility, and the reliability of the archives.

VIII. Data Processing and Delivery

As the mission was evolving, the needs of the clients were more clearly defined, and requirements were made with more refinements. A major upgrade of the ScanSAR processor was completed in 2002. The upgrade produces a significant improvement in image quality and radiometry [6,8]. In addition, wide bandwidth communication links made processing and delivery feasible within delays that were considered impossible at the beginning of the mission.

At the Canadian Data Processing Facility (CDPF), products are generated according to three levels of priority: REGULAR–data are processed and delivered within the next 14 days using the definitive orbit parameters; RUSH–data are processed and delivered within 48 h after data downlink, and Near Real Time (NRT)–data are processed and delivered within 4 h after data downlink. Because of the increased processing speed of new processors and the availability of high-speed Internet link, data can be processed and delivered to clients well within the prescribed delays (NRT processing delay is currently under an hour with a 4-h requirement). Electronic delivery through high-speed Internet link is now preferred by a majority of customers. Furthermore, when low-loss compression

algorithm is applied to the image products, the transmission time can be further reduced.

IX. Data Quality

Using an Amazon and Boreal forest calibration site, and the RADARSAT-1 precision transponders, radiometry and spatial accuracy of the data are routinely monitored and maintained [3]. Point target measurements indicate that the image quality is consistently maintained to a level still better than required by the original system specification. All beams are now fully calibrated. When required, re-calibration is performed to accommodate changes in SAR antenna elevation beam pattern. For completeness, high-incidence beams (EH3, EH4, and EH6) were calibrated in February 2005. ScanSAR image quality significantly improved with the upgrade of the CDPF processor in 2002. The upgrade eliminated processing problems such as scalloping, location error due to pulse repetition frequency (PRF) ambiguity, visibility of beam boundaries, automatic gain control (AGC), and saturation error.

X. Upcoming Challenges

Originally built for five years, the systems were never meant to be operational after 11 years. Some of the applications still used are not supported anymore, and it would be too expensive to rebuild them from scratch. Databases are getting to their limits, systems architectures were not meant to accommodate so many requests, some fields in databases were not meant to hold seven digits, etc. Many of these issues have been solved, but we are expecting more. Funding is also an issue, as long-term operations were not expected.

On the other hand, earlier this year, MDA GSI informed CSA that several data reception facilities would like to join the network of RADARSAT-1 certified stations. This is quite an honor for the system, but considering the depleting resources, this will be another challenge.

With the upcoming launch of RADARSAT-2, RADARSAT-1's role is bound to change. After the commissioning of RADARSAT-2, the current operational mandate will be changed to a supporting mandate for RADARSAT-2 operations. In addition, discussions are under way to see how RADARSAT-1 could be used to support data acquisitions over the poles during the International Polar Year 2007–2009.

Conclusion

After completing over 10 years of operations, the RADARSAT-1 system is more then ever providing quality data to the user community. Many modifications have been made to the system and still other improvements are under way. Since the beginning of the mission, the Mission Management Office was able to prepare all the acquisition plans without missing a single one, and the overall efficiency of the RADARSAT-1 system is better than 95.5%. Although RADARSAT-1 was built for five years of operations, data quality is still better than original system requirements due to routine monitoring of radiometric and spatial positioning of the data.

Improvements made in the last few years have increased the speed, accessibility, stability, reliability, security, maintainability, and upgradability of the system. The users can benefit from turnaround time that was unthinkable in the early days of the mission. While the system is aging gracefully, many challenges are still on the way.

References

[1] Parashar, S., RADARSAT-1 Program Overview, "RADARSAT-1 Extended Mission Review," Canadian Space Agency, 29–30 April, 2002.

[2] Mahmood, A., Background Mission, "RADARSAT-1 Extended Mission Review," Canadian Space Agency, 29–30 April, 2002.

[3] Srivastava, S., User Demand and Service Delivery, "RADARSAT-1 Extended Mission Review," Canadian Space Agency, 29–30 April, 2002.

[4] Showalter, D., System Operations and Satellite Status, "RADARSAT-1 Extended Mission Review," Canadian Space Agency, 29–30 April, 2002.

[5] Srivastava, S. K., "RADARSAT-1 Mission Management Office (MMO) Operation," *SpaceOps 2004 Conference*, 17–21 May 2004, Montreal, Quebec, Canada; also http://www.spaceops2004.com/downloads/papers/SPACE2004sp-template00320F.pdf.

[6] Cote, S., Srivastava, S. K., Le Dantec, P., Hawkins, R. K., Murnaghan, K. "RADARSAT-1 Image Quality Evolution to the Extended Mission," *SpaceOps 2004 Conference*, 17–21 May 2004, Montreal, Quebec, Canada; http://www.spaceops2004.com/downloads/papers/SPACE2004sp-template00203F.pdf.

[7] McNally, J., Monson, R., and Koeppen, J., "RADARSAT Data Policy," Canadian Space Agency, RSCSA-PR-0004, 1994.

[8] Weisz, S., Benalia, A., Banik, B., Truong, Y., Marchand, D. "RADARSAT-1 Mission Planning System Evolution," Poster presented at *SpaceOps 2004 Conference*, 17–21 May 2004, Montreal, Quebec, Canada.

Index

This page intentionally left blank

Color Figures

Chapter 3

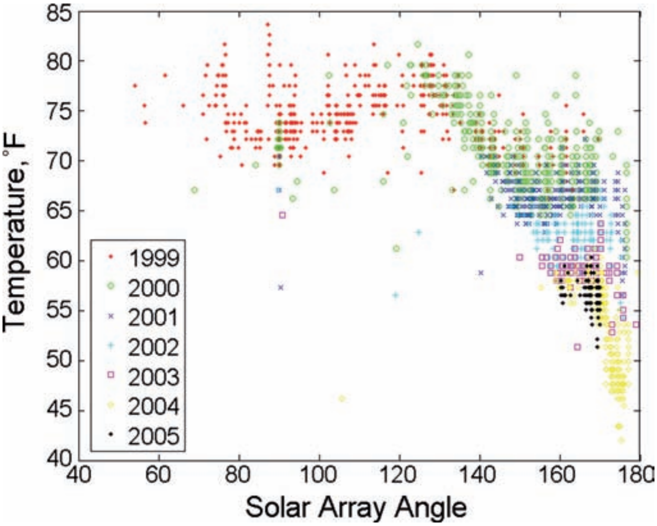


Fig. 9 Thermistor B temperature at time of heater turn-on vs solar array angle.

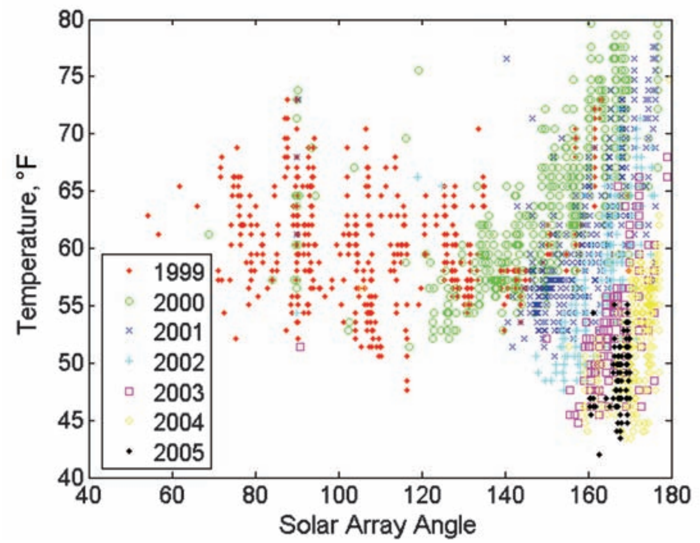


Fig. 11 Thermistor C temperature at time of heater turn-on vs solar array angle.

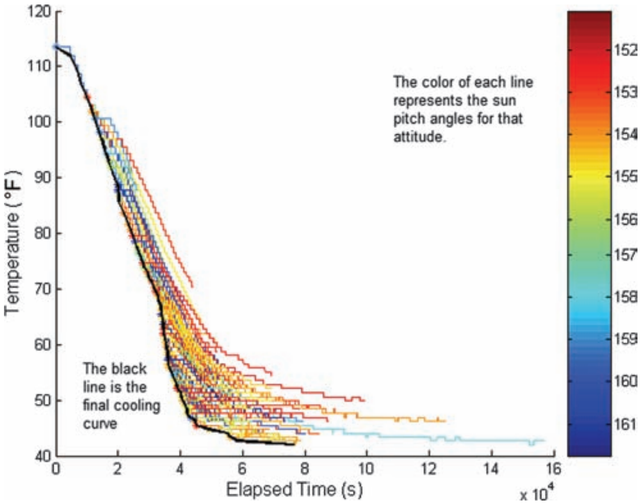


Fig. 13 Final cooling curve.

Chapter 10

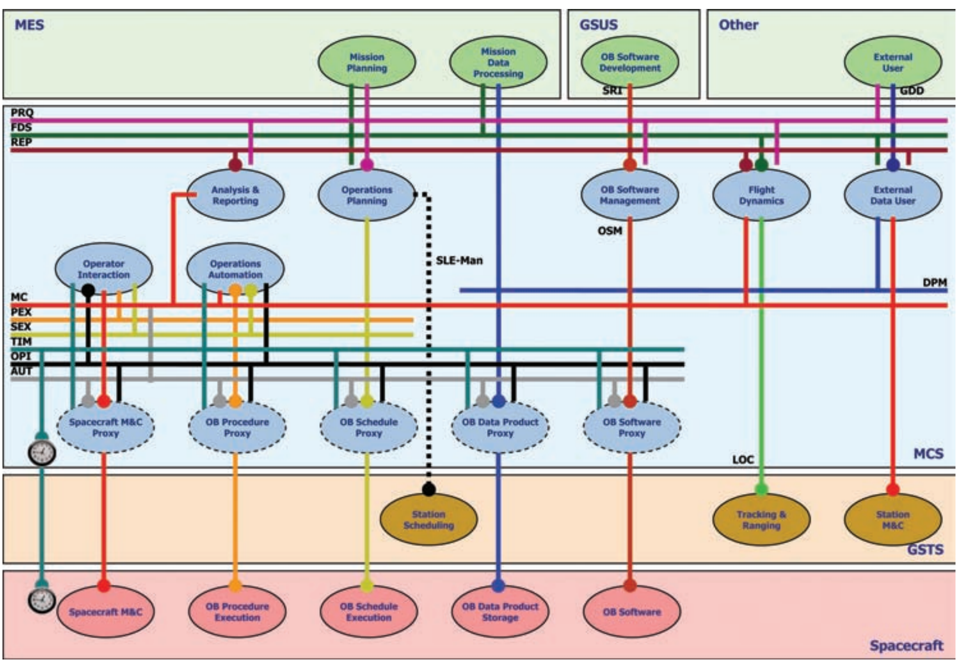


Fig. 5 GDSS mission operations services.

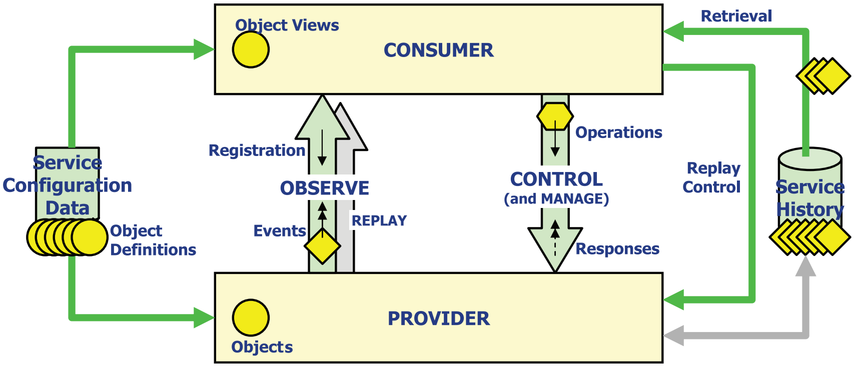


Fig. 9 Operation interaction pattern.

Chapter 12

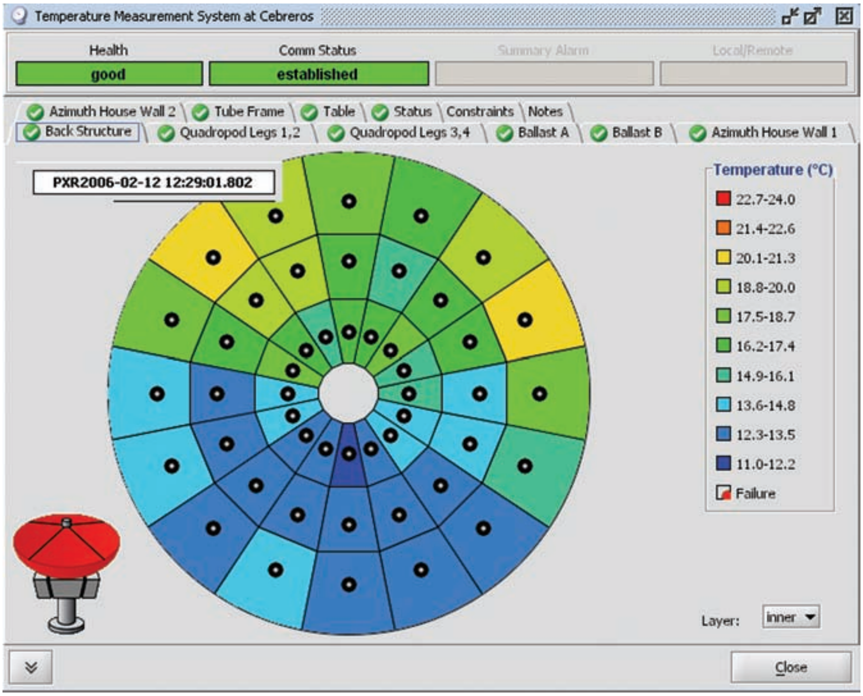


Fig. 4 Screen capture of PCS TMS display window.

Chapter 15

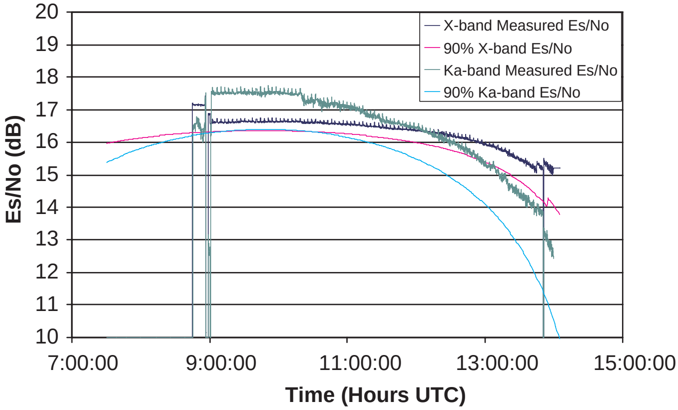


Fig. 8 Day 05-360 DSS-34 Ka-band and X-band E_s/N_0.

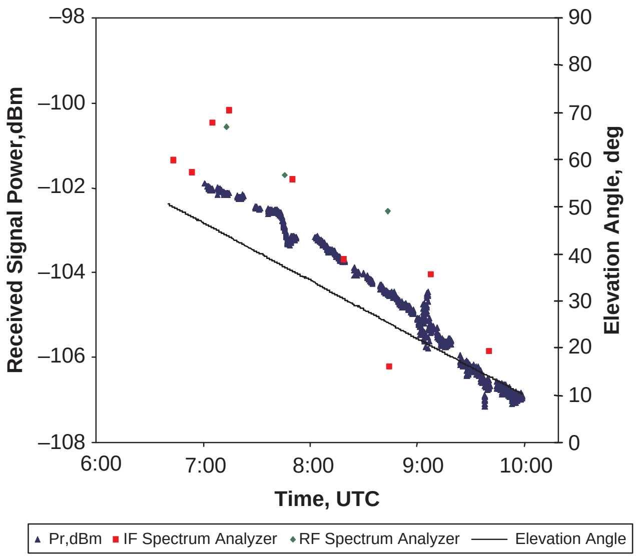


Fig. 10 Measured Ka-band signal strength received at DSS-13 on day 05-350.

Chapter 16

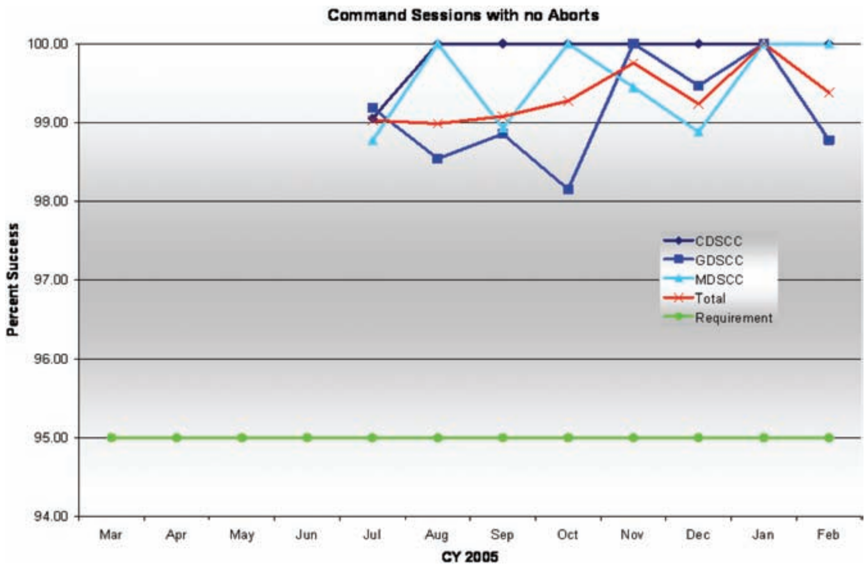


Fig. 6 Figure of merit for command quantity/quality metrics.

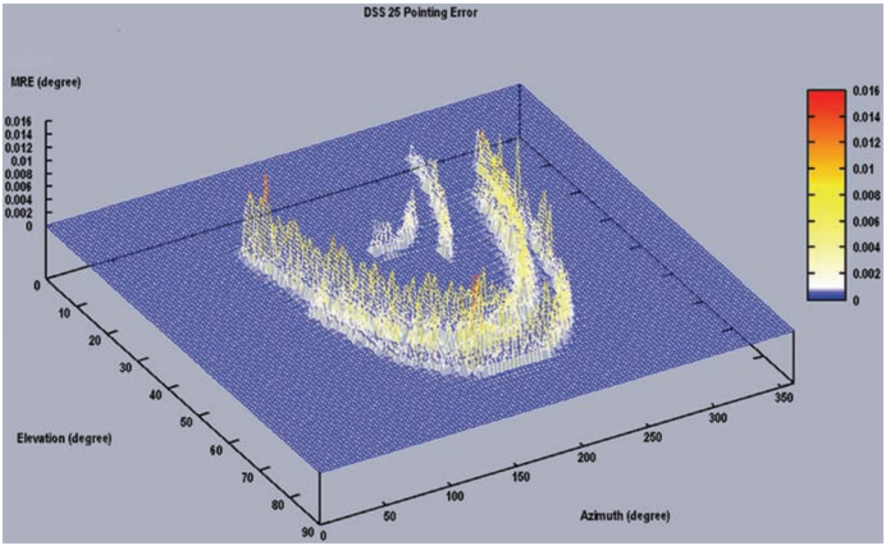


Fig. 10 Sample pointing accuracy of a 34-m beam waveguide antenna.

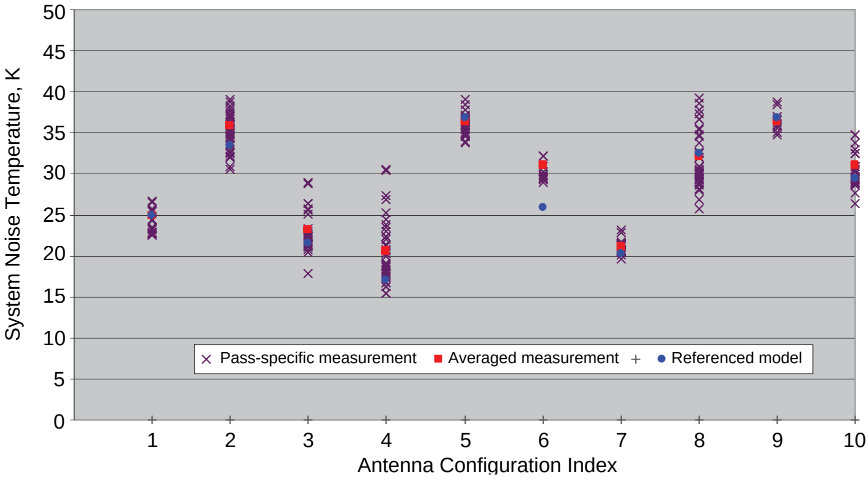


Fig. 11 Sample of observed system noise temperatures at Goldstone (near zenith).

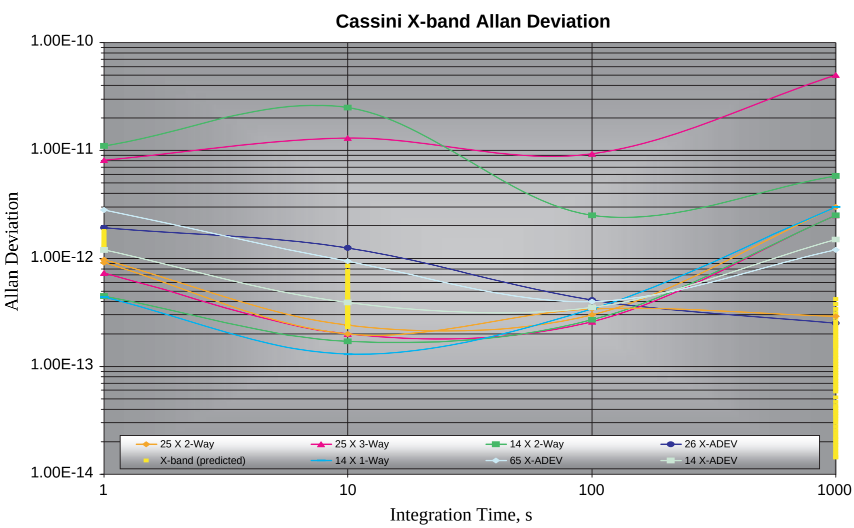


Fig. 13 Frequency stability between observed and model.

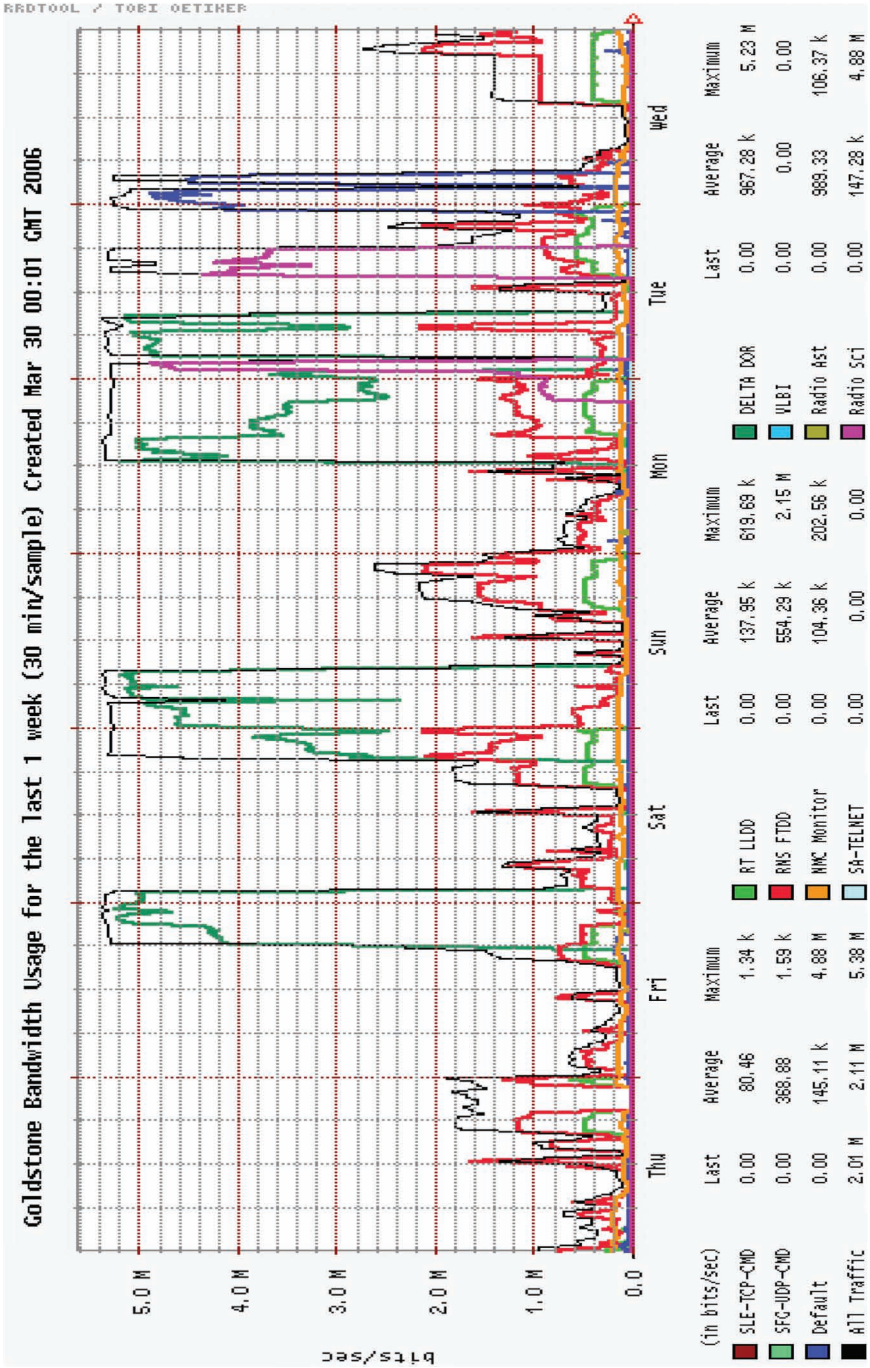


Fig. 15 Sample of network utilization.

Chapter 17

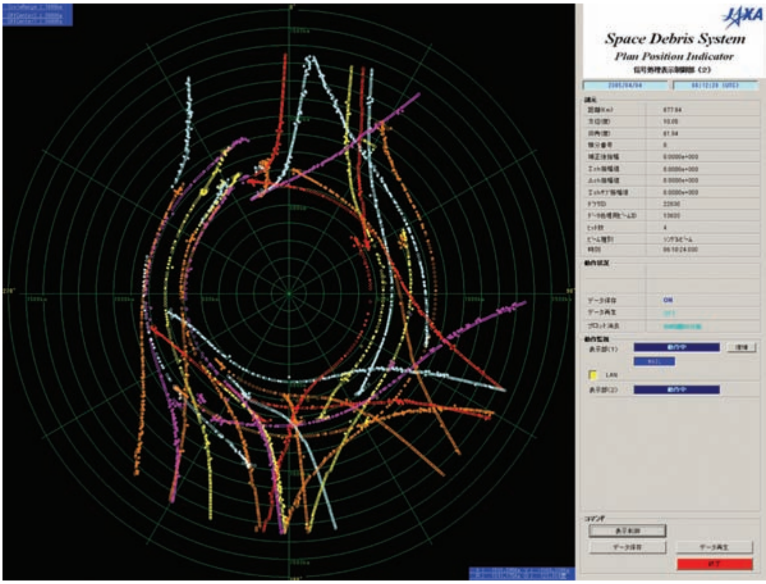


Fig. 2 Plan position indicator at TKSC. Each line represents each trajectory of one space debris object.

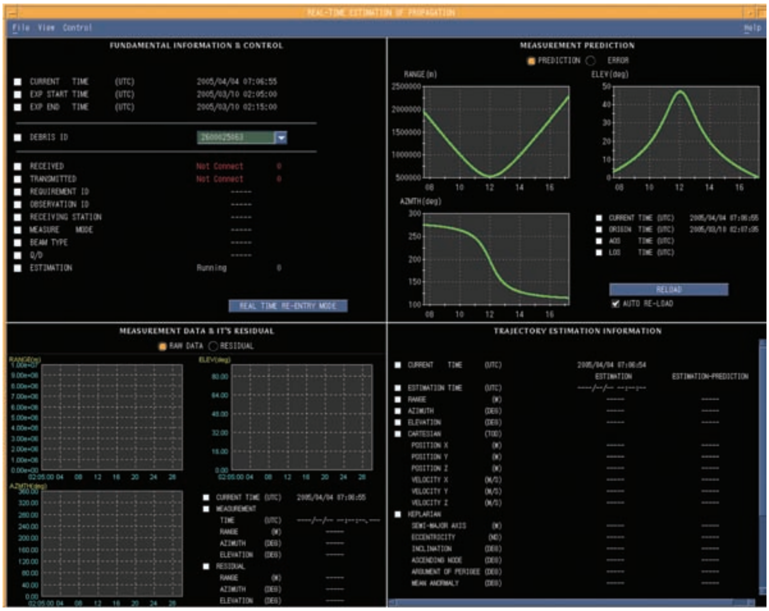


Fig. 3 Real-time trajectory estimation program. The green line represents the predicted azimuth/elevation/range. Once the real-time trajectory estimation runs, a red line will be shown in the same fields.

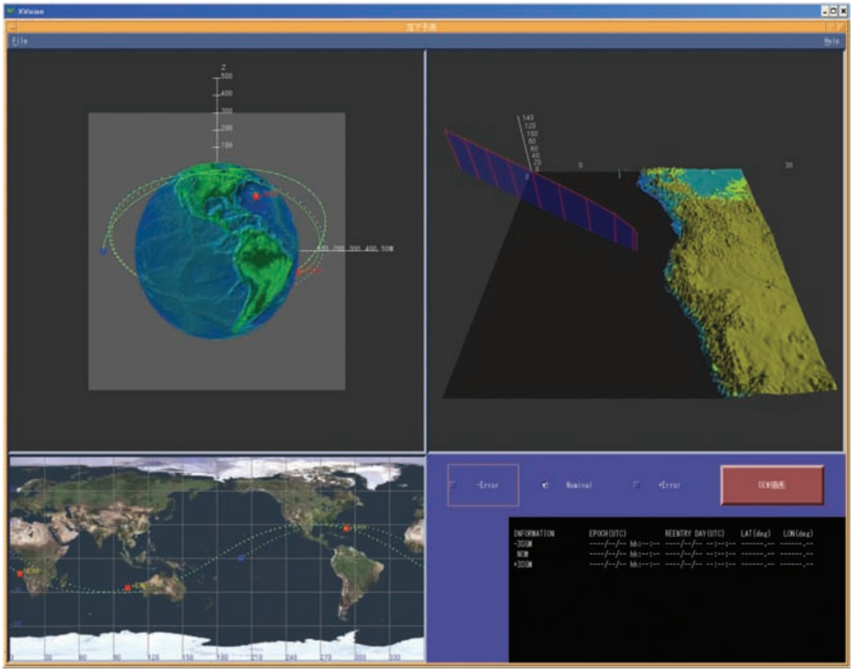


Fig. 4 Screen image of reentry prediction. The predicted reentry area (nominal and ± 20% prediction errors are included) is shown in the two-dimensional and three-dimensional world map.

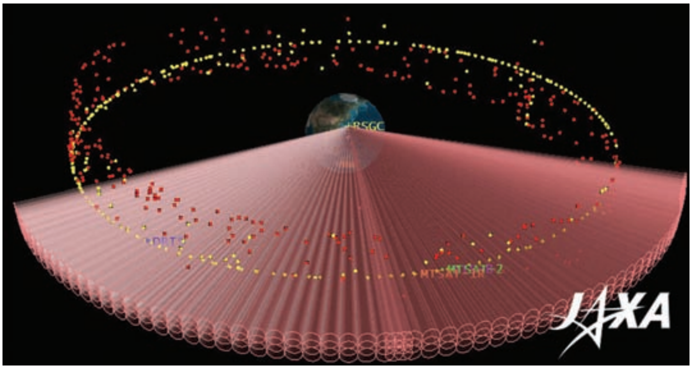


Fig. 5 JAXA GEO-belt survey. Yellow dots are operational satellites and red dots are space debris. One pink colored circle represents a field of view for the 1-m telescope. The survey fields are separated into 120.

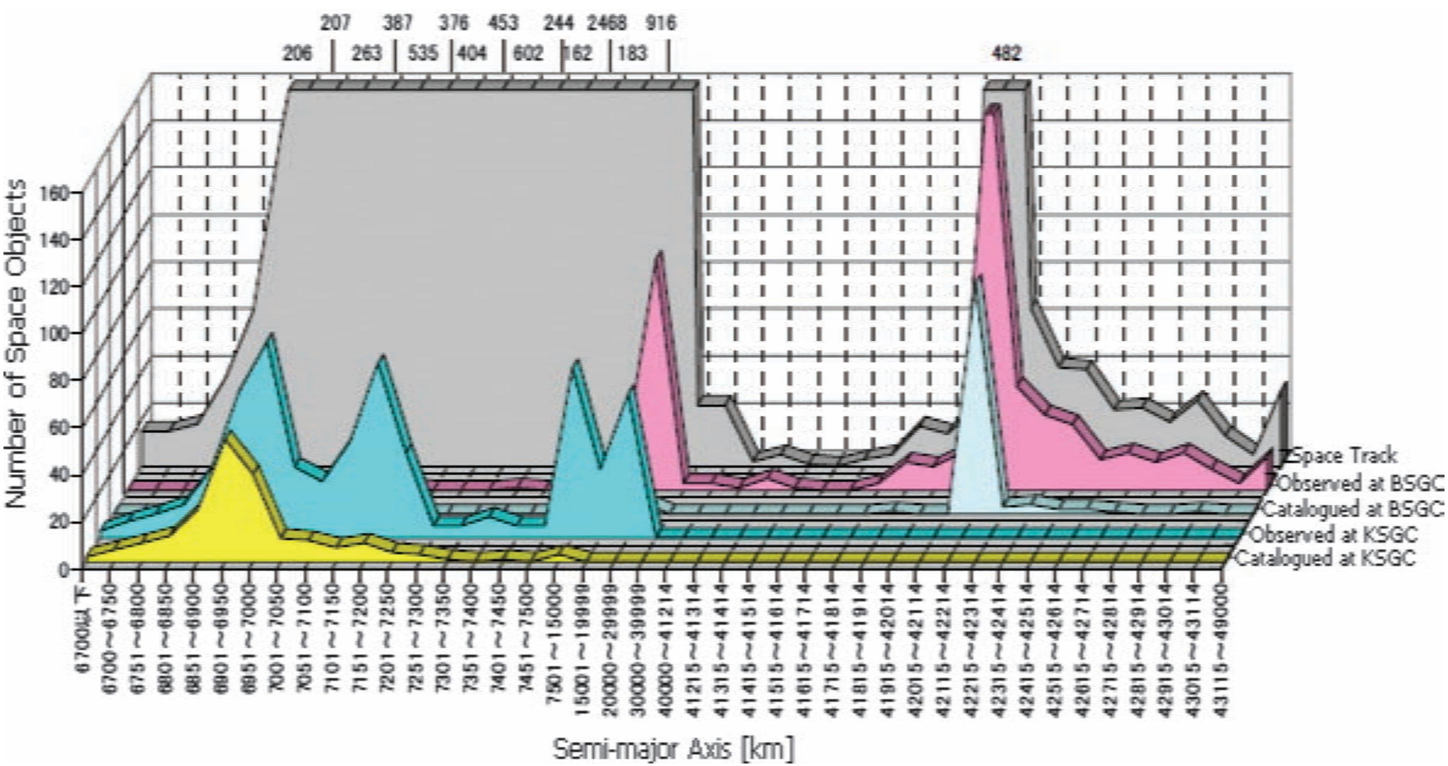


Fig. 6 Distribution of observed/cataloged objects (as of 31 May 2006): the yellow block, objects cataloged at KSGC; sky blue block, observed at KSGC; aqua block, cataloged at BSGC; pink block, observed at BSGC; and gray block, catalogued by Space Track.

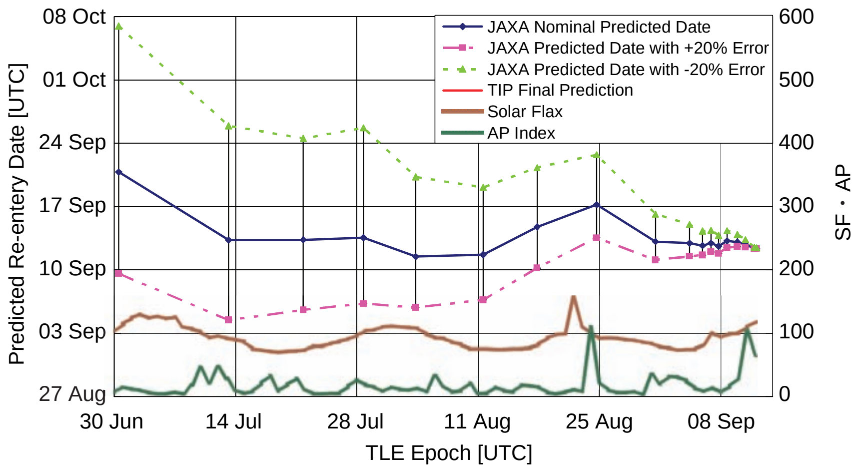


Fig. 10 Predicted date of YOHKOH's reentry.

Chapter 19

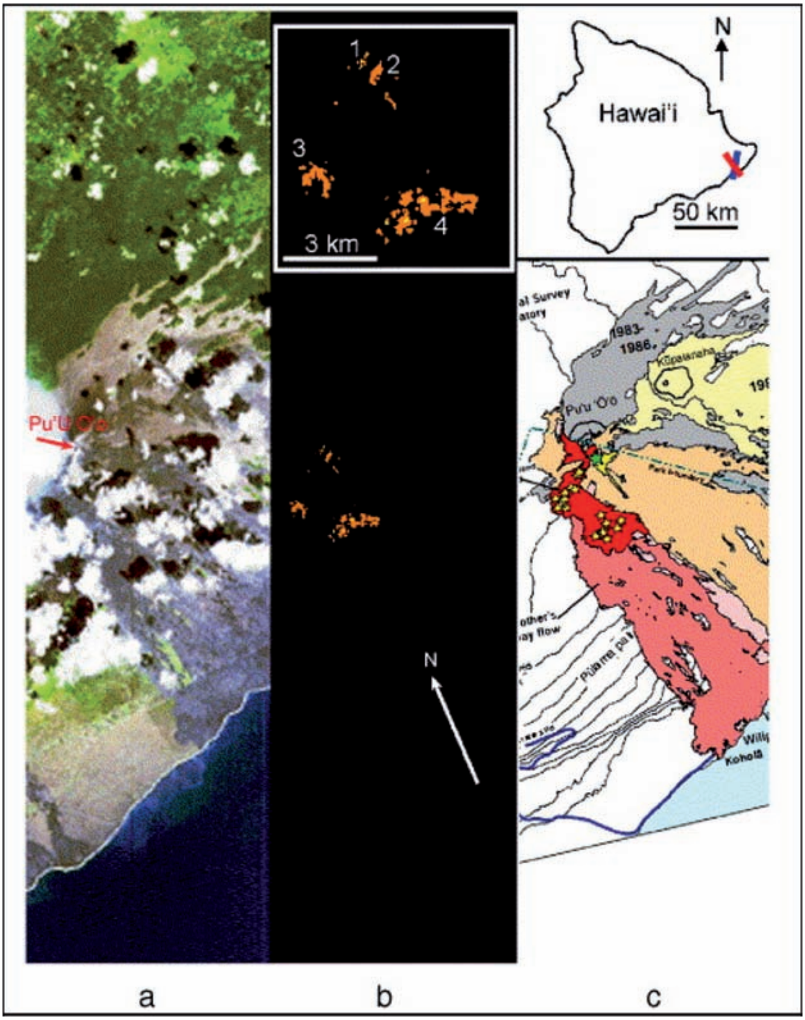


Fig. 1 Kilauea Volcano: a) the visible image of Kilauea, Hawaii, on 24 January 2004; b) thermal classifier output including an inset enlargement of the active area; c) the USGS – Hawaiian Volcano Observatory map showing volcanically active areas in January 2004. Yellow areas delineate the Martin Luther King (MLK) flows in January 2004.

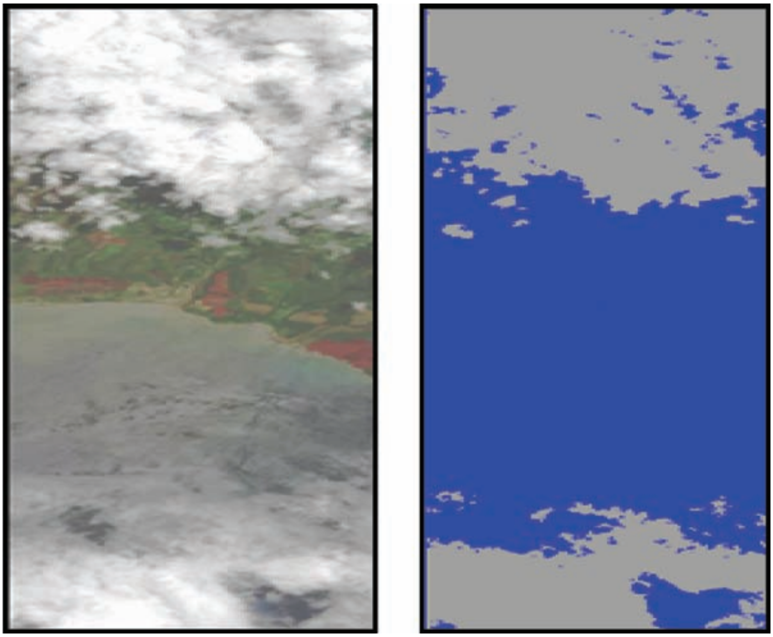


Fig. 2 Cloud detection—visual image at left, grey in right image indicates detected cloud.

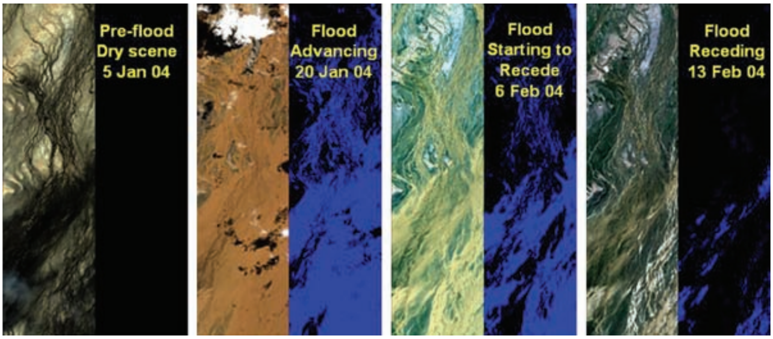


Fig. 3 Flood detection time series imagery of Australia's Diamantina River with visual spectra at left and flood detection map at right. Flooding is caused by monsoonal rain.

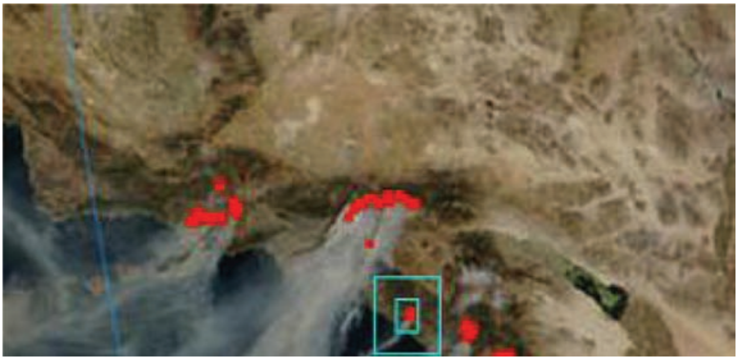


Fig. 6 Active fire alerts for the October 2003 Southern California fires. Red indicates active fires. The light blue box illustrates the background region used in the relative threshold detection.

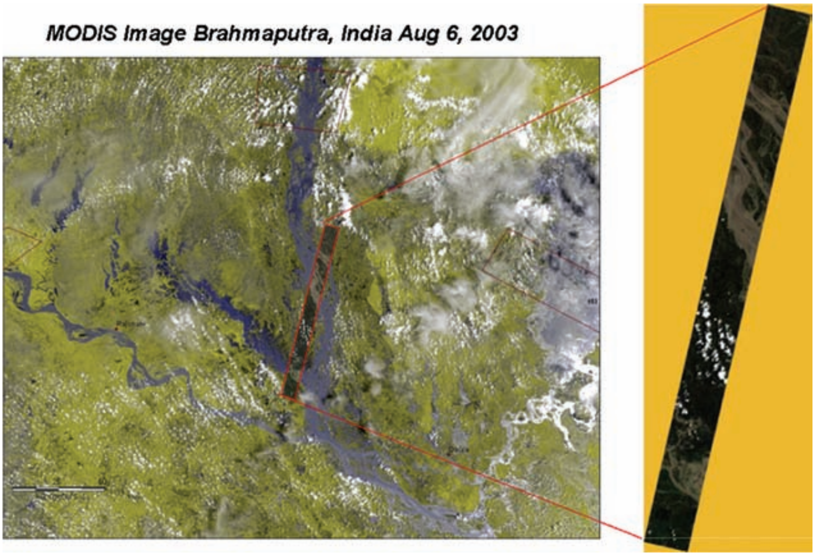


Fig. 7 Examples of 250-m low-resolution MODIS imagery (left) and 30-m EO-1 imagery (right) from the flood sensorweb capturing Brahmaputra River flooding in India, August 2003.

Chapter 29

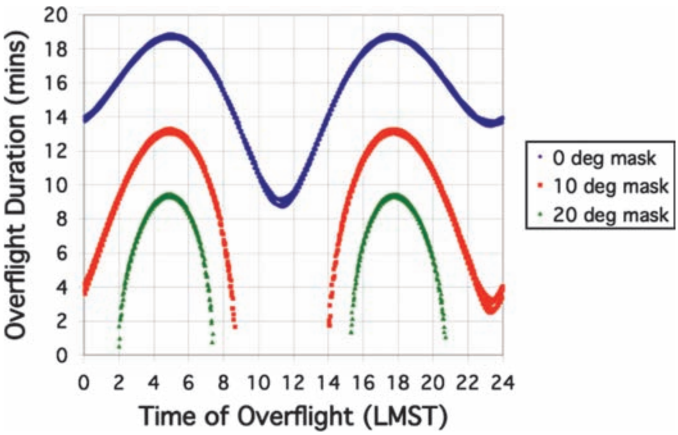


Fig. 2 Odyssey overflight duration vs local solar time, 67.5°N latitude.

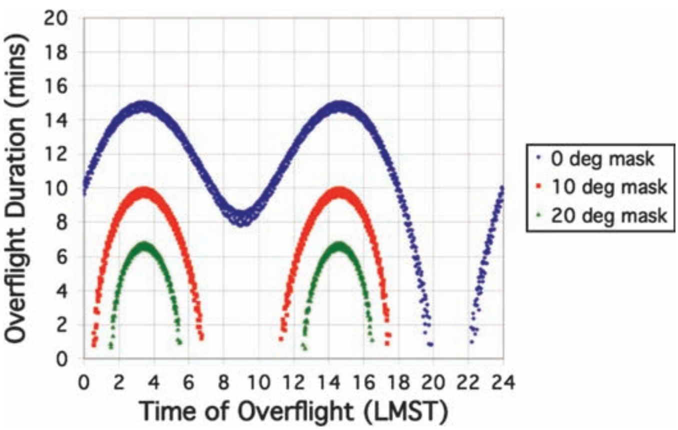


Fig. 3 MRO overflight duration vs local solar time 67.5°N latitude.

Chapter 32

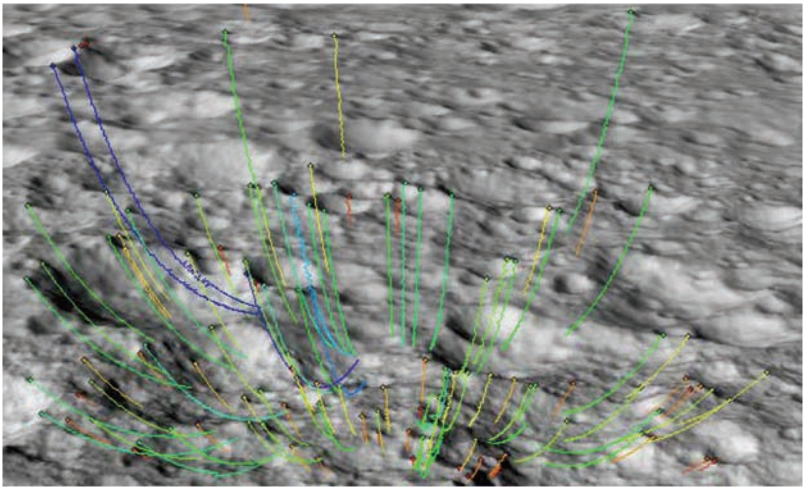


Fig. 3 Tracking features in a PANGU image sequence.

Author Index

Supporting Materials

A complete listing of titles in Progress in Astronautics and Aeronautics and other AIAA publications is available at http://www.aiaa.org.